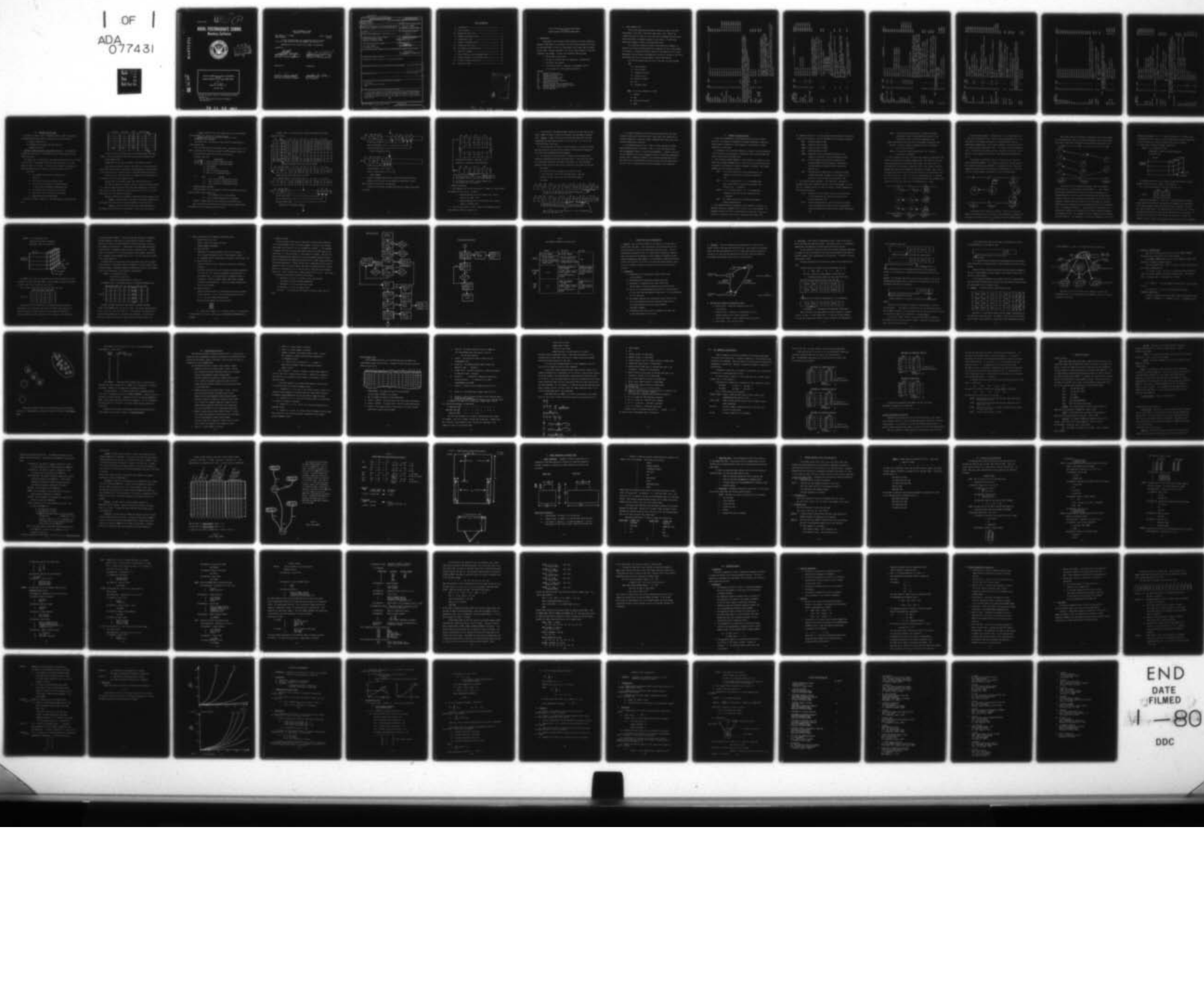
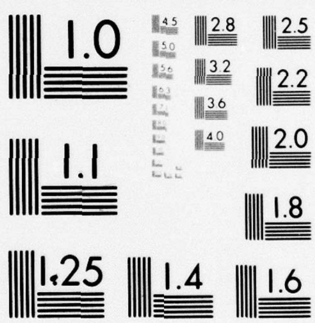
AD-A077 431 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 15/7
TACTICAL PARAMETERS AND INPUT REQUIREMENTS FOR THE GROUND COMPO--ETC(U)
OCT 79 S H PARRY , E P KELLEHER MIPR-CD-1-79
UNCLASSIFIED NPS55-79-023 NL

1 OF 1
ADA
077431



END
DATE
FILMED
1-80
DDC



MICROCOPY RESOLUTION TEST CHART

LEVEL III

NPS55-79-023

NAVAL POSTGRADUATE SCHOOL

Monterey, California

TACTICAL PARAMETERS AND INPUT REQUIREMENTS
FOR THE
GROUND COMPONENT OF THE STAR COMBAT MODEL

by

Sam H. Parry
Edward P. Kelleher, Jr.

October 1979

Approved for public release; distribution unlimited.

Prepared for:
The U.S. Army Training & Doctrine Command
Fort Monroe, VA.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral T. F. Dedman
Superintendent

Jack R. Borsting
Provost

The work reported herein was accomplished with the support of the U.S. Army Training & Doctrine Command, Fort Monroe, VA.

Prepared by:

S H Parry
Sam H. Parry
Department of Operations Research

Edward P. Kelleher
Edward P. Kelleher, Jr.
Department of Operations Research

Reviewed by:

Michael G. Sovereign, Chairman
Department of Operations Research

Released by:

William M. Tolles
William M. Tolles
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 14 NPS55-79-023

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER: 9

4. TITLE (and Subtitle): 6 Tactical Parameters and Input Requirements for the Ground Component of the STAR Combat Model.

5. TYPE OF REPORT & PERIOD COVERED: Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): 10 Sam H. Parry ~~and~~ Edward P. Kelleher, Jr.

8. CONTRACT OR GRANT NUMBER(s): 12 95

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 15 MIPR-CD-1-79

11. CONTROLLING OFFICE NAME AND ADDRESS:
U.S. Army TRADOC
Fort Monroe, VA 23651

12. REPORT DATE: 11 Oct~~ober~~ 1979

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Simulation of Tactical Alternative Responses

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Land Combat
Tactical Decision Algorithms
STAR

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report provides a description of each of the functional modules of the Ground Component of the STAR Combined Arms Combat Simulation Model. A complete description of each input variable is provided in the order in which they are read by the model. This report is one of a series of STAR publications.

251 450

JOB

TABLE OF CONTENTS

I.	Introduction	1
II.	Input Variables	2
III.	Movement Decision Logic	9
IV.	Movement Coordination Logic	17
V.	Counterattack Logic	30
VI.	Sequence of Movement Areas	36
VII.	Target Selection Tactics	39
VIII.	XM-1 Ammunition Redistribution	45
IX.	Ammunition Resupply	49
X.	Target Dimensions and System/Ammo Codes	57
XI.	Defender Movement to Full Defilade Tactics	60
XII.	Accuracy/Lethality Data Arrays	62
XIII.	Suppression Model	73

Tactical Parameters and Input Requirements for the Ground Component of the STAR Combat Model

I. INTRODUCTION

The purpose of this volume of the Simulation of Tactical Alternative Responses (STAR) Model is to provide a list of all input variables and arrays for the ground model, as well as a description (and in many cases an example) of each input. Section 2 provides a list of all input variables in the order required by the program. For each, the following items are given:

a. The variable or array name

b. The type of variable (and array dimension, if appropriate)

c. The mode of the variable

d. A brief description or reference to a subsequent section in the volume for a more complete description.

Sections

III.	Movement Decision Logic
IV.	Movement Coordination Logic
V.	Counterattack Logic
VI.	Sequence of Movement Areas
VII.	Target Selection Tactics
VIII.	XM-1 Ammunition Redistribution
IX.	Ammunition Resupply
X.	Target Dimensions and System/Ammo Codes
XI.	Defender Movement to Full Defilade Tactics
XII.	Accuracy/Lethality Data Arrays
XIII.	Suppression Model

II. INPUT VARIABLE LIST

This section describes each variable required as input to the STAR Ground Model in the order in which each data item is read. Many of the variables are part of a specific logic block (such as Movement Decision Logic) and are described in subsequent sections of this Volume. In those cases appropriate reference is made in the Description Field.

This Volume has combined a narrative description with examples, in addition to the usual variable definition, to enhance the user's understanding of the model. It should be noted that several of the modules are still under development (such as Resupply and Suppression, as well as the play of the "Dirty Battlefield) and will be fully documented in future STAR reports.

The following abbreviations are used in the table for TYPE and MODE.

<u>TYPE</u>

GV: Global Variable

LV: Local Variable

TA: Temporary Attribute

TE: Temporary Entity

RV: Recursive Variable

SET: Sets

PE: Permanent Entity

<u>MODE</u> - If an array, dimension is noted.

I: Integer

R: Real

RD: Double Precision Real

A: Alpha

Variable or ARRAY Name	Type	Mode	Description	Routine
OFFLOAD	GV	I	Section IX	MAIN
LOADUP	GV	I	Section IX	MAIN
TOW1CASE	GV	I	Section IX	MAIN
TOW2CASE	GV	I	Section IX	MAIN
DRAGON	GV	I	Section IX	MAIN
PCF1,...,PCF6	GV	I	Section IX	PILE.SO.CREATE
NUM.PILES	GV	I	Section IX	PILE.SO.CREATE
N.SUPPLY.OFFICER	GV	I	Section IX	PILE.SO.CREATE
D1,...,D6	TA	I	Section IX	PILE.SO.CREATE
LOCATION	TA	I	Section IX	PILE.SO.CREATE
GFAREA	TA	I	Section IX	PILE.SO.CREATE
MY.SO	TA	I	Section IX	PILE.SO.CREATE
ATP.AREA	TA	I	Section IX	PILE.SO.CREATE
PLACES	GV	I(2-D)	Section IX	PILE.SO.CREATE

The next set of input variables is read by ROUTINE RESTER and is prepared as a package to set up battlefield coordinates, macro terrain and all geometric overlays (such as forests) that directly enter the LOS computation. For example, concealment features are handled probabilistically and therefore do not directly enter LOS computations.
(See Reference 1 for description of Terrain/LOS Model).

Variable or ARRAY Name	Type	Mode	Description	Routine
NUMBER.OF.SYSTEMS	GV	I	Section X	MAIN
J,L	RV	I	Section X	MAIN
TARDIM	GV	R(3-D)	Section X	MAIN
WH.1,...,WH.5 } WM.1,...,WM.3 }	GV	I	Section XI	MAIN
TTIME	GV	R	This real variable is used to activate a debugging traceback feature built into numerous STAR events and routines. If activation is not required, read in a value of TTIME which is larger than the value used to stop the simulation (SIM.STOP). If SIM.STOP is read as 4500.0, TTIME could appear on a data card as: 5000. This will prevent the debugging traceback routine from being activated. Normal system tracebacks are still available with no action required by the user.	MAIN

Variable or ARRAY Name	Type	Mode	Description	Routine
NUM.DS.ARRAYS	RV	I	Section VII	DANGER.STATE
WD	RV	I	Section VII	DANGER.STATE
SYSTYPE	RV	I	Section VII and Section X	DANGER.STATE
WPNTYPE	RV	I	Section VII and Section X	DANGER.STATE
POINT.HOLD	GV	I(2-D)	Section VII	DANGER.STATE
N	RV	I	Section VII	DANGER.STATE
ARRAY(thru DATA)	PA	I(2-D)	Section VII	DANGER.STATE
COM.VAL	GV	R	A real variable input as a value between 0.0 and 1.0 representing the probability of communicating with platoon members during execution of platoon fire coordination logic. Example of input is shown below: .75 (communication allowed in 75% of the attempts).	MAIN
NCASE	GV	I	A user defined integer run number from 0 to 999.	MAIN
CASE	GV	I	Assumes integer values 1 through 4: 1 = 40 round XM1/120mm 2 = 43 round XM1/120mm 3 = 49 round XM1/120mm 4 = 52 round XM1/120mm or 55 round XM1/105 These values are used by routine RELOAD to take appropriate ammunition redistribution action.	MAIN
ONDISK	GV	I	1∼Write output of shot list and final attribute list to user specified disk files 0∼Paper output only	MAIN
XY CARDS	GV	I	1∼Record the X-Y coordinates, name, alive-dead status, system type and weapon type of each element in the battle every POS.PLOT units. 0∼Otherwise	MAIN
ABORTPRINT	GV	I	1∼Tally and print at the end of the simulation, the number and reasons for aborted fire events 0∼No tally and print.	MAIN

Variable or ARRAY Name	Type	Mode	Description	Routine
NPRI	RV	I	Section III	MAIN
TABLE(thru TEMPL)	PA	I(3-D)	Section III	MAIN
RPRI	RV	I	Section III	MAIN
RTAB(thru TEMPL)	PA	I(3-D)	Section III	MAIN
BLCOMP	GV	I	Number of defender maneuver commanders	MAIN
BBN	GV	I	Number of defender battalion commanders	MAIN
RBN	GV	I	Number of attacker battalion commanders	MAIN
PHZLINES	GV	I	See Section IV	ARRAY.CORD.SET
PHSCORD	PA	I(2-D)	See Section IV	ARRAY.CORD.SET
BNCORD	PA	I(3-D)	See Section IV	ARRAY.CORD.SET
COCORD	PA	I(3-D)	See Section IV	ARRAY.CORD.SET
BN.AIR.PRI	PA	I(2-D)	Defined in Air-Air Defense Volume	ARRAY.CORD.SET
MU	GV	R(2-D)	Array MU contains a parameter of the normal distribution associated with the AMSAA supplied log-normally distributed load times for various vehicle types as a function of range to the target. Row 1 is for all tanks, Row 2 is for all other weapon types. Each value represents a 500 meter band starting with the 0-500 meter band.	MAIN
R.NUM.ALIVE	GV	I	Total number of attacker elements to be created	MAIN
B.NUM.ALIVE	GV	I	Total number of defender elements to be created.	MAIN
DELTA.T	GV	R	Time interval between STEP.TIME events in simulated seconds	MAIN
PNUM	GV	I	Total number of PLATOON.LEADER entities to be created	MAIN
CNUM	GV	I	Total number of COMPANY.COMMANDER entities to be created	MAIN
NOTE:	The following variables input in Routine RES.MOV are described in detail in Volume V-STAR Movement Model			
ID	RV	I	PLATOON NUMBER	RES.MOVE
NE	RV	I	No. Elements in Platoon ID	RES.MOVE
NA	RV	I	No. Areas used by Platoon ID	RES.MOVE

Variable or ARRAY Name	Type	Mode	Description	Routine
POSITION	GV	R(3-D)	Position in Platoon ID	RES.MOVE
ID	RV	I	Platoon Number	RES.MOVE
NM	RV	I	No. Routes used by Platoon ID	RES.MOVE
MOVE.DATA	GV	I(2-D)	Area/Route Numbers for Platoon ID	RES.MOVE
N	RV	I	Number of Routes to use	RES.MOVE
ID	RV	I	Route Number	RES.MOVE
NM	RV	I	Number of Movement Control Points on Route	RES.MOVE
ROUTE.DATA	GV	R(2-D)	Coordinates for each MCP and Formation Code to be used	RES.MOVE
N	RV	I	No. of Movement Formations	RES.MOVE
ID	RV	I	Formation Number	RES.MOVE
NM	RV	I	No. Positions in Formation ID	RES.MOVE
FORM.OFFSET	GV	R(2-D)	ΔX, ΔY Offsets for each position in formation ID	RES.MOVE
SIZE	RV	I	Section VI	ASSIGN.ORDERS
SEQUENCE.OF.AREAS	PA	I(1-D)	Section VI	ASSIGN.ORDERS
R.PCT.ATT	GV	R	A real number between 0.0 and 1.0 representing the fraction of attacker losses which when reached or exceeded will stop the simulation.	MAIN
B.PCT.ATT	GV	R	Similar to R.PCT.ATT except for defender forces	MAIN
POS.PLOT	GV	I	Time Interval at which Position Plots made (see XY CARDS)	MAIN
SIM.STOP	GV	I	Time at which simulation will stop	MAIN
SEED.V	PA	I(1-D)	Uniform Random Number Seeds	MAIN
	The next set of input data relates to accuracy and lethality data arrays from Routines RES.2 through RES.5. These data arrays are defined in Section XII in the required order.			
CDTIME	GV	R	Section VIII	MAIN
BTIME	GV	R	Section VIII	MAIN

Variable or ARRAY Name	Type	Mode	Description	Routine
S1TIME	GV	R	Section VIII	MAIN
S2TIME	GV	R	Section VIII	MAIN
CASEAP	GV	I	Section VIII	MAIN
CASEHE	GV	I	Section VIII	MAIN
CAPDS	GV	I	Section VIII	MAIN
CHEAT	GV	I	Section VIII	MAIN
CDA	GV	I	Section VIII	MAIN
CDH	GV	I	Section VIII	MAIN
BA	GV	I	Section VIII	MAIN
BH	GV	I	Section VIII	MAIN
S1A	GV	I	Section VIII	MAIN
S1H	GV	I	Section VIII	MAIN
S2A	GV	I	Section VIII	MAIN
S2H	GV	I	Section VIII	MAIN
BLUTNUM	GV	I	Total Number of Times Defender Forces are to be created	MAIN
BLUGUYS	GV	I	Number of Defender elements to be created for this creation	MAIN
T.BLUGUYS	GV	I	Time to next creation of defender elements	MAIN
MOV.BLU	GV	I	{ 1 create elements in moving state { 0 otherwise	MAIN

Note: The following variables are the initial attribute list for all Blue (Defender) elements to be created at the start of the battle (one data card per element).

Variable or ARRAY Name	Type	Mode	Description	Routine
NAME	TA	I	Element Number	BL.CREATE
SYS.TYPE	TA	I	System Code (See Section X)	BL.CREATE
WPN.TYPE	TA	I	Weapon Code (See Section X)	BL.CREATE
SEC	TA	I	Section Number	BL.CREATE

Variable or ARRAY Name	Type	Mode	Description	Routine
PLT	TA	I	Platoon Number	BL.CREATE
CO	TA	I	Company Number	BL.CREATE
BN	TA	I	Battalion Number	BL.CREATE
SECLDR	TA	I	Section Leader Element Number	BL.CREATE
PLTLDR	TA	I	Platoon Leader Element Number	BL.CREATE
COCDR	TA	I	Company Commander Element Number	BL.CREATE
AREA.START	TA	I	Starting Movement Area Number - Section IV	BL.CREATE
NOTE:	This ends defender element data.			
N.ECHELONS	GV	I	Total Number of Attacker Echelons to be created during battle	MAIN
N.CURR.ECH	GV	I	Current Echelon Number	MAIN
T.CUR.ECH	GV	I	Time at which current Echeon is created	MAIN
MOV.ECH	GV	I	1 Create Elements in moving state 0 Otherwise	MAIN
Note:	The nine variables which make up the attacker element attribute list (analog to defender attribute list except that SEC and SECLDR are omitted) are input at this point for all attacker elements to be created in the battle.			
CRITICAL.VALUE	GV	R	Percent of target height visible, less than which LOS does not exist	MAIN
THUP	GV	R	Time to detect a firing target, given looking in 30 degree	MAIN
THLO	GV	R	sector at fire,is U(THLO,THUP)	MAIN
NIUP	GV	R	Time to detect a firing target, given looking in 90 degree	
NILO	GV	R	sector at fire,is U(NILO,NIUP)	MAIN
DATA.SP	PA	I(2-D)	Section XIII	SET.SP
VALUU	RV	R	Section XIII	SET.SP
N.CAS	GV	I	Section V	SET.CA
ENTRY	RV	I	Section V	SET.CA
CA.DATA	PA	I(2-D)	Section V	SET.CA

III. MOVEMENT DECISION LOGIC

1. The desire to move a unit or individual vehicle in STAR is represented by movement decision logic which is based on two distinct criteria:

   1) Range to enemy force, or

   2) Combination of friendly attrition level and Red/Blue force ratio.

2. Defender's Movement Decision Logic (Range Criteria). To implement the movement decision logic for the defender, the user inputs values into the global array Table.

The Table array is 3-dimensional, the dimensions of which are 7 by (4+3*NPRI) by M where NPRI is the number of attrition level actions and M is the number of planes of the array. (M must be at least equal to the number of defending maneuver units, a maneuver unit being a company sized unit).

User Inputs:

1) Sys.type/Wpn.type he wants to monitor, up to a total of 6.
2) The range at which each system will move.
3) The 0/1 value indicating if a monitored system is restricted (see Movement Coordination Logic).
4) The range at which the entire unit will move.
5) Range within which Force Ratio is calculated.
6) Range beyond which nothing happens.

The first 4 columns of TABLE for a particular maneuver unit might look like:

Sys.Type	Wpn.Type	Range	Restricted?	
1	1	800	1	
1	2	800	1	
2	3	1000	0	
2	4	1200	0	
3	6	0	0	
5	1	1200	0	
3000	2500	800	1	

The first 6 rows of the TABLE array are information about individual systems. The 7th row contains information about the entire maneuver unit.

In the example array:

1st row refers to Sys.Type 1, Wpn.Type 1 which might be an XM1.

The user has specified in Column 4 that XM1's are a restricted system (which means permission must be granted by higher headquarters before any XM1's in this maneuver unit are allowed to move to subsequent defensive positions).

Column 3 of row 1 specifies that if the enemy closes to 800 meters of the maneuver until XM1's will request to move.

3rd row refers to Sys.Type 2, Wpn. Type 3 which might be IFV's.

The user has specified in Column 4 that IFV's are not restricted systems (i.e., they may move without permission from higher headquarters), and if the enemy closes to within 1000 meters of the maneuver unit all IFV's in the unit will move to their next defensive positions.

7th row is reserved for information concerning the entire maneuver unit:

Column 1 specifies that if the enemy is not within 3000 meters of the maneuver unit, the unit will not move regardless of attrition level.

Column 2 specifies that all enemy units within 2500 meters of the maneuver unit constitute the red elements in force ratio calculations.

Column 3 specifies that if the enemy closes to within 800 meters of the maneuver unit, the unit will request to move.

Column 4 specifies unit must have permission to move.

3. Allowable Actions (Defender)

The actions that a user desires to take place are represented by a 2 digit action code.

The first digit specifies the system that is to take the action (1-6). Where 1 indicates the 1st monitored system 2 the 2nd monitored system, etc.

The second digit specifies the action to take place.

Allowable codes are:

	Second Digit	
1st digit may be 1 to 6	- 1	move a section of monitored system
	- 2	move a platoon of monitored system
	71	move Company
	72	move Battalion
	81	move all un-monitored systems
	82	mount-up all dismounted elements

E.G.	Code	Action
	11	move a section of 1st monitored system
	21	move a section of 2nd monitored system
	32	move a platoon of 3rd monitored system

4. Allowable Actions (Attacker)

Attacking maneuver units are generally defined as battalion sized units. Actions allowed for the attacker are:

11 Place maneuver unit in hasty defense

12 Cause attacking maneuver unit to withdraw

5. Defender's Movement Decision Logic (Attrition Level/Force Ratio).

User inputs attrition level/force ratio/action codes by weapon system and cumulative total in decending attrition levels.

Columns 5 thru M of the array for a particular maneuver unit might appear as follows:

Sys	Wpn	%	FR	Act	%	FR	Act	%	FR	Act	%	FR	Act	
1	1	75	3	72	75	2	71	50	2	82	50	2	22	99
1	2	50	3	71	50	2	32	50	2	32	25	2	82	99
2	3	50	3	32	50	2	31	25	3	42	25	2	41	99
2	4	50	3	42	50	2	41	99						
3	6	0	99											
5	1	50	2	62	99									
3000	2500	75	3	71	50	2	42	50	1	82	99			

Consider the second row of the array as follows:

Sys	Wpn	Rng	Rest?	%	FR	Act	%	FR	Act	%	FR	Act	%	FR	Act	
1	2	800	1	50	3	71	50	2	32	50	2	32	25	2	82	99

Assume that Sys.Type 1/Wpn.Type 2 is an XM1

	Sys	Wpn	Rng	Rest?	%	FR	Act	%	FR	Act	%	FR	Act	%	FR	Act	
XM1	1	2	800	1	50	3	71	50	2	32	50	2	32	25	2	82	99

	Sys	Wpn	Rng	Rest?		%	FR	Act	
XM1	1	2	800	1		25	2	82	99

At 25% loss of XM1's

and if Forces Ratio $\geq$ 2 to 1

Action 82 takes place

(Mount-up all Dismounted Elements)

	Sys	Wpn	Rng	Rest?		%	FR	Act	%	FR	Act	
XM1	1	2	800	1		50	2	32	50	2	32	

At 50% loss of XM1's

and if Force Ratio ≥ 2 to 1

Action 32 takes place

(Move 2 platoons of IFV's)

	Sys	Wpn	Rng	Rest?	%	FR	Act	
XM1	1	2	800	1	50	3	71	

At 50% loss of XM1's

and Force Ratio ≥ 3 to 1

Action 71 takes place

(Move Company)

6. Attacker's Movement Decision Logic (Attrition Level/Force Ratio).

Movement Decision Logic for the attacker is simplified in that it doesn't take force ratios into consideration.

A plane of the attacker's Movement Decision Array (RTAB) might appear as follows:

Sys	Wpn	%	Act	%	Act	
1	7	75	12	50	11	99
2	8	90	12	75	11	99
0	0	75	11	99		

Assume that Sys.Type 1 , Wpn.Type 7 = T72.

Looking at a single row of the RTAB array:

	Sys	Wpn	%	Act	%	Act	
T72	1	7	75	12	50	11	99

At 50% attrition of T72's in this maneuver unit the maneuver unit will go into a Hasty Defense.

At 75% attrition of T72's in this maneuver unit the maneuver unit will withdraw.

7. Comments and Cautions

a. Every Maneuver unit on the battlefield is linked to a single plane of the movement decision array. In general:

1) Defending Maneuver Units are company sized and are linked to the TABLE array.

2) Attacking Maneuver Units are battalion sized and are linked to the RTAB array.

b. The user must input at least 1 plane of the movement decision array (TABLE or RTAB) per maneuver unit.

c. The structure of the RTAB and TABLE arrays are such that the user may input a number of planes in either array greater than the number of maneuver units. <u>However</u>, if done, the user must also code the logic which tells the program when and under what conditions a particular unit is to start using a different plane of the array.

d. If the user does not desire to use Force Ratio as a criteria for movement, the insertion of a zero in the FR columns of the array will cause attrition level alone to determine desired movement.

e. The insertion of an action code 13 instructs the simulation to do nothing at a particular attrition level/Force Ratios. Since attrition levels for a particular system are continuously calculated (i.e., not reset when movement occurs) the use of the 13 action code may be very useful to the user.

For example:

For simplicity assume a maneuver unit consists of only 5 XMI's and XM1's are the 1^{st} monitored system; the user may wish to input a row of the Table Array which appears as follows:

Sys	Wpn	Rng	Rest?	%	FR	Act	%	FR	Act	%	FR	Act	%	FR	Act	
1	2	800	1	80	2	71	60	2	71	40	0	13	20	2	71	99

The loss of the 1^{st} XM1 will cause the maneuver unit to move to its 2^{nd} defensive position

Sys	Wpn	Rng	Rest?	%	FR	Act	%	FR	Act	%	FR	Act	%	FR	Act	
1	2	800	1	80	2	71	60	2	71	40	0	13	20	2	71	99

The maneuver unit is now on its 2^{nd} Defensive position, the loss of a 2^{nd} XM1 causes <u>no</u> action to take place

- but -

when a 3^{rd} XM1 (out of the original 5) is lost the maneuver unit will move again, etc.

f. It should be apparent from the previous discussions that the input provided by the user for attrition levels, force ratios, and ranges must be carefully thought out so that the scheme of maneuver which takes place in the simulation makes sense tactically.

g. The Movement Decision Logic in STAR is closely related to the Movement Coordination Logic of the model. That is, a unit that wants to move must be permitted to move by a controlling headquarters. Likewise a unit that has not requested to move may be ordered to move by a controlling headquarters based on what is happening elsewhere in the controlling headquarters sector. The interaction of the coordination and decision logic must be understood by the user if he is to get the desired tactical movement in the simulation.

IV. MOVEMENT COORDINATION LOGIC

1. During the implementation of STAR Movement Decision Logic it may be desirable to coordinate the movements of Company and Battalion sized units based on what is happening to other Companies in the Battalion or other Battalions in the Brigade.

To accomplish this coordination STAR uses a system of "tactical weighting" of unit positions, and a concept of "coordination lines" which delineate the trace of units along the Brigade front.

2. Every Defending Company Commander in the simulation is created as a permament entity (Company.Commander) and is filed in a set called Battalion which corresponds to the battalion to which the Company is assigned. Each Company. Commander has the following attributes:

COWT - The tactical weight (or relative importance) of the Company's position along a particular coordination line.

CMSN - A 0/1 attribute which is 0 if the Company does <u>not</u> have permission to move off a particular coordination line, 1 otherwise.

CREQST - A 0/1 attribute which is 0 if the Company has requested to move from current coordination line, 1 otherwise.

COMPY - An integer attribute which indicates the Company's number (identification).

3. Every Defending Battalion Commander in the simulation is created as a permanent entity (Bn. Commander) and is filed in a set called Brigade. Bn. Commander entities are filed in the Brigade set if and only if they are to occupy the coordination line currently occupied by the Brigade, thus,

Bn. Commander entities are filed and removed from the Brigade set during the course of the simulation. Each Bn. Commander has the following attributes:

BNWT - Battalion analog to COWT

BMSN - Battalion analog to CMSN

BREQST - Battalion analog to CREQST

BATT - Battalion analog to COMPY

BNCUR - Represents the sum of the COWT attributes of each Company in the battalion which has requested to move.

BNLO - The lower bound on BNCUR, which if equaled or exceeded by BNCUR causes each Company in the Battalion to have permission to move from their current coordination line.

BNGO - The upper bound or BNCUR which if equaled or exceeded by BNCUR constitutes an <u>order</u> for each Company in the Battalion to move to their next coordination line.

4. The Defending Brigade Commander in the simulation is created as a permanent entity (Bde. Commander) and owns a set called Brigade in or from which Bn. Commander entities are filed or removed. The Bde. Commander has the following attributes:

BDECUR - Represents the sum of BNWT attributes of each Battalion in the Brigade which has requested to move from the current coordination line.

BDELO - The lower bound on BDECUR, which if equaled or exceeded causes each Battalion currently filed in the Brigade set to be given permission to move from the current coordination line.

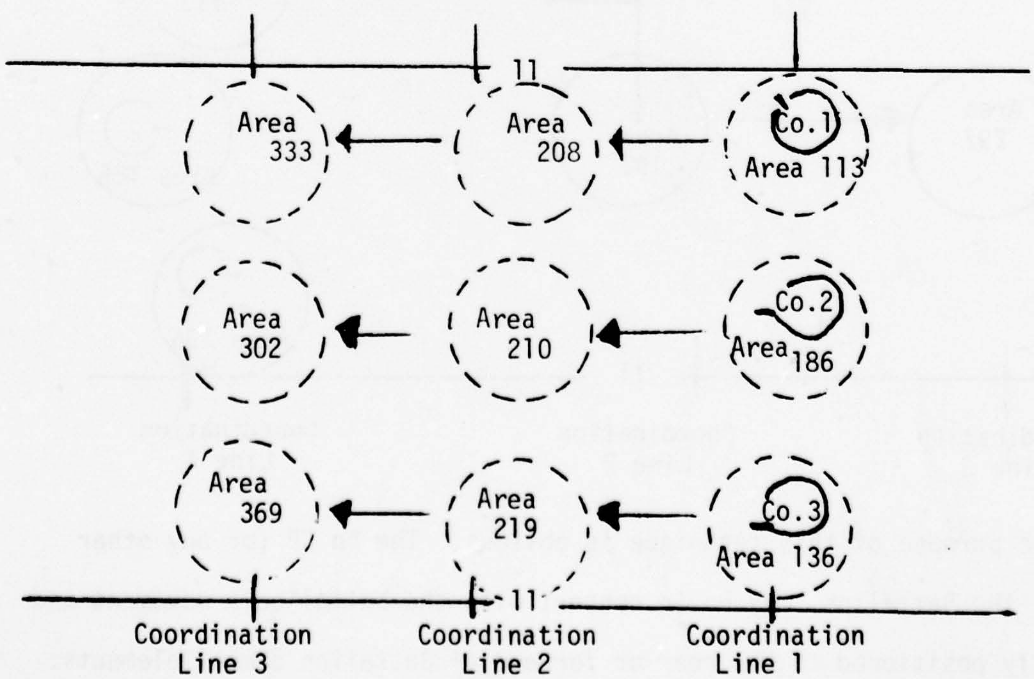
BDEGO - The upper bound on BDECUR which if equaled or exceeded constitutes an order for each Battalion currently in the Brigade set to move to their next coordination line.

Global variables which are utilized in the movement coordination logic:

PHZLINES - the total number of coordination lines for defending units in the simulation.

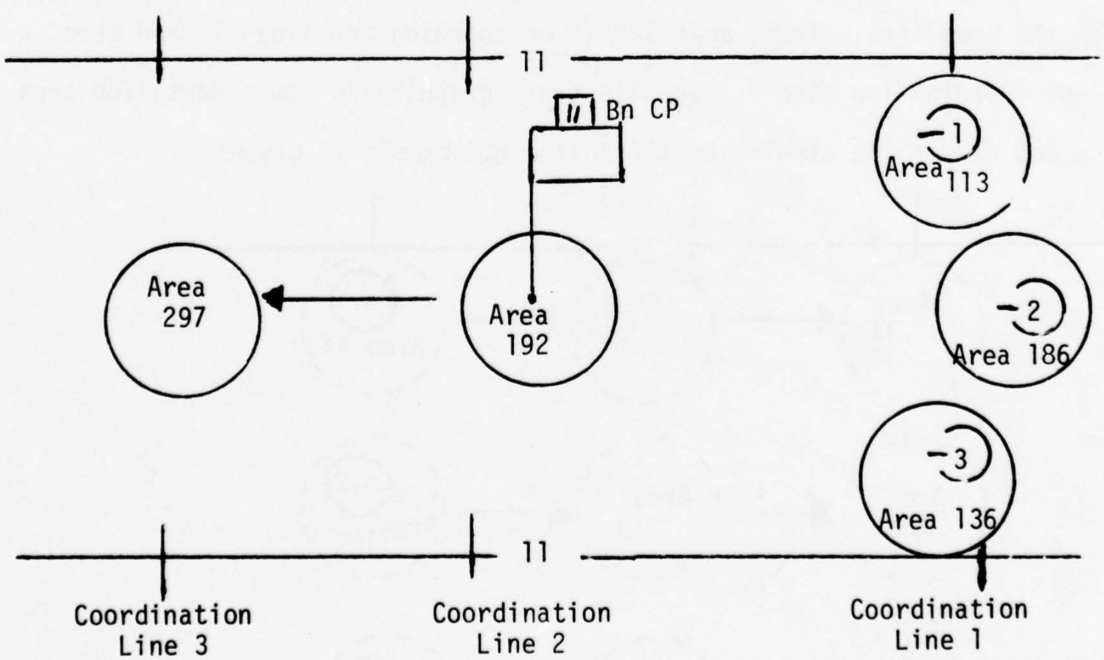
PHAROW - the currently occupied coordination line. (Currently occupied by the Brigade).

5. It may be useful at this point to discuss the concept of coordination lines and their relationship with Company movement areas. Each entity in the simulation which represents an individual vehicle (tank, ITV, Truck, etc.) has associated with it an attribute called Area. Start. Area. Start is the unit movement area from which the element's next move will be made. Area. Start is an integer attribute, the first digit of which specifies the coordination line on which the area lies. Thus, area 123 is on coordination line 1 and area 263 is on coordination line 2. To illustrate graphically how a Battalion area might be set up for the simulation the following example is given:



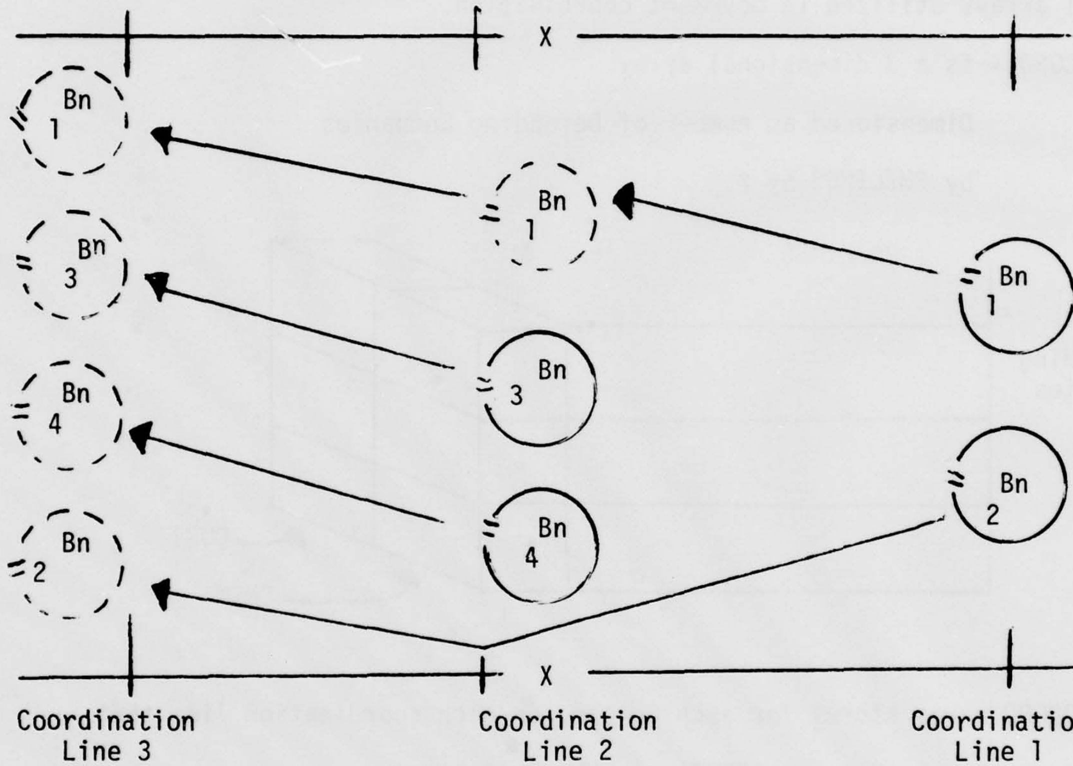
As shown above, Company 1 occupies area 113 on coordination line 1, area 208 on coordination line 2 and area 333 on coordination line 3, and each vehicle in the Company would have on its attribute Area. Start the appropriately numbered area. Thus a tank in Company 1 would have Area.Start = 113 while occupying coordination line 1 or in transit back to coordination line 2. Upon arrival at coordination line 2 the tank's Area.Start is set to 208, etc. Other Companies and vehicles in the simulation are handled similarly.

The concept of coordination lines is a construct of the simulation used to coordinate movement. It is important to note that an area need not fall directly on what has been defined as a coordination line. A Battalion command post could occupy an area coded as on coordination line 1, but physically at or behind another coordination line as illustrated below.



The purpose of this technique is obvious. The Bn CP (or any other element of the Battalion) may be in concert with the battalion's movement and continuously positioned to the rear or forward of Battalion combat elements.

We can now extend our discussion of coordination lines to the Brigade as illustrated below: (Keep in mind that the Companies within each Bn would have areas assigned which correspond to the appropriate coordination line).



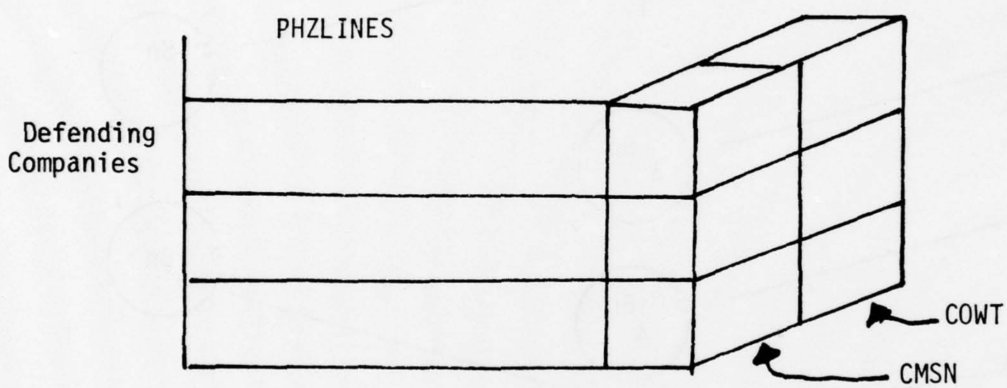
At the start of this Simulation Battalions 1 and 2 are occupying coordination line 1, and the global variable PHAROW = 1 . Bn. Commander entities which correspond to Bns 1 and 2 are filed in the Brigade set. Although Battalions 3 and 4 are physically located on the battlefield and allowed to fight from their positions on coordination line 2, their corresponding Bn. Commander entities are <u>not</u> filed in the Brigade set, and these Battalions are not allowed to move to subsequent positions until coordination line 1 is evacuated and PHAROW = 2. Upon evacuation of coordination line 1 (evacuation is defined as all elements which are able to move have departed the coordination line) Bn. Commander (1) and Bn. Commander (2) are removed from the Brigade set. Bn. Commander 1, 3, and 4 are subsequently filed in the

Brigade set and PHAROW is set = 2. Note that Battalion 2 is allowed to move to coordination line 3, but will not be allowed to move from that coordination line until actions have caused coordination line 2 to be evacuated.

6. Global arrays utilized in movement coordination.

COCORD - is a 3 dimensional array

Dimensioned as number of Defending Companies by PHZLINES by 2.



The COCORD array stores for each Company on each coordination line that Company's relative importance (COWT) to the Battalion.

For a single coordination line the array might look like:

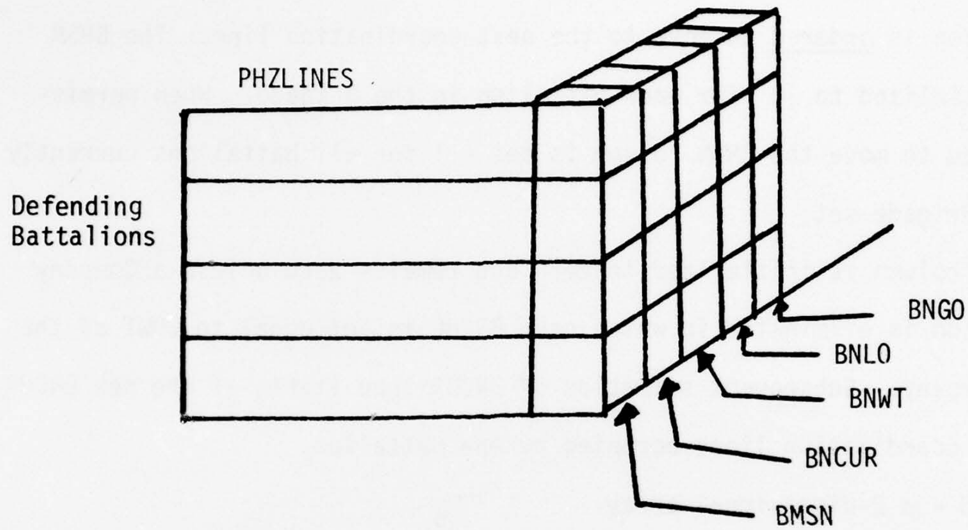
		CMSN	COWT
Company	1	0	1
	2	0	3
	3	0	1

Assuming that Companies 1, 2, and 3 are in the same Battalion, the COWT Column represents the relative importance of each Company to the maintenance of an effective defense along the coordination line in question. The user must input the COWT values. The CMSN values are initialized to zero indicating no Company has permission to move from the coordination line in question. At some point in the battle the battalion may give permission, or order, the Companies to move, in which case the CMSN column will be set=1 for all Companies.

BNCORD - is a 3-dimensional array

Dimensioned as number of Defending

Battalions (BBN) by PHZLINES by 5.



The BNCORD array stores for each Battalion on each coordination line that battalion's relative importance (BNWT) to the Brigade, the upper bound (BNLO) on BNCUR. BNWT, BNLO, and BNGO are user input.

For a single coordination line the array might look like:

		BMSN	BNCUR	BNWT	BNLO	BNGO
	1	0	0	1	2	3
	2	0	0	2	1	4
Battalion	3	0	0	3	1	2
	4	0	0	1	1	2

Assuming that all four battalions are occupying positions along the coordination line in question, the values in the BNWT column indicate the relative importance of each battalion in the Brigade in maintaining an effective defense along the coordination line in question. The BNLO column

is the lower bound on BNCUR. If the sum of the COWT attributes of Companies who have requested to move equals or exceeds BNLO the battalion requests permission to move, and if given permission transmits permission to move to each Company in the battalion. If BNCUR equals or exceeds BNGO each Company in the battalion is <u>ordered</u> to move to the next coordination line. The BMSN column is initialized to 0 for each battalion in the Brigade. When permission is granted to move the BMSN column is set = 1 for all battalions currently filed in the Brigade set.

The BNCUR column is initialized to zero and remains zero unless a Company in the battalion is eliminated in which case BNCUR is set equal to COWT of the eliminated Company. Subsequent summation of BNCUR then starts at the new BNCUR value for all coordination lines occupied by the battalion.

PHSCORD - a 2-dimensional array

Dimensioned as PHZLINES by

(number of Defending Battalions (BBN) + 3)

The PHSCORD array is the Brigade analog to the COCORD and BNCORD arrays.

PHZLINES	BDECUR	BDELO	BDEGO	Bn 1	Bn 2	Bn 3	Bn(n-1)	Bn(n)
1	0	3	5	/	/	/	/	/
2	0	2	3	/	/	/	/	/
3	0	3	6	/	/	/	/	/
4	0	3	4	/	/	/	/	/

The BDELO and BDEGO columns are used similar to the BNLO and BNGO columns of the BNCORD array, and are user inputs. The columns that correspond to Bn 1, Bn 2, etc. are 0/1 values; 1 if a battalion is occupying or en-route to the coordination line in question zero otherwise. The use of the Bn 1-Bn(n) columns are discussed in the discussion of Routine Coord. Set.

7. Events and Routines used in Movement Coordination Logic:

1) Routine Array. Cord. Set:

Reads in user input values into arrays COCORD, BNCORD, and PHSCORD.

2) Event Phaz.Chk:

Determines if current coordination line is still occupied. If still occupied reschedules itself, if not occupied, removes Bn. Commander entities from Brigade set and calls routine Coord. Set.

3) Routine Coord. Set:

Called initially out of B1. Create or by Phaz.chk as simulation continues.

Determines which battalions are to occupy the current coordination line, sets the 0/1 values corresponding to the Battalions in the PHSCORD array. Files appropriate Bn. Commander entities in Brigade Set. Initializes BNCUR, BREQST, BNWT, BNLO, and BNGO attributes for each battalion. Initializes CREQST and COWT attributes for each Company.

4) Routine Decision

Called by routine Action or by routine Bug.Chk returns a variable (Order) which says 1 = yes = you can perform desired action or 0 = no = you can't perform desired action.

Routine Decision updates the following values:

CREQST
BNCUR
BREQST
BDECUR

It causes entire Companies in a battalion to move or all battalions in the Brigade set to move if upper bounds on BNCUR/BDECUR (BNGO/BDEGO) are exceeded.

8. Requests to Move

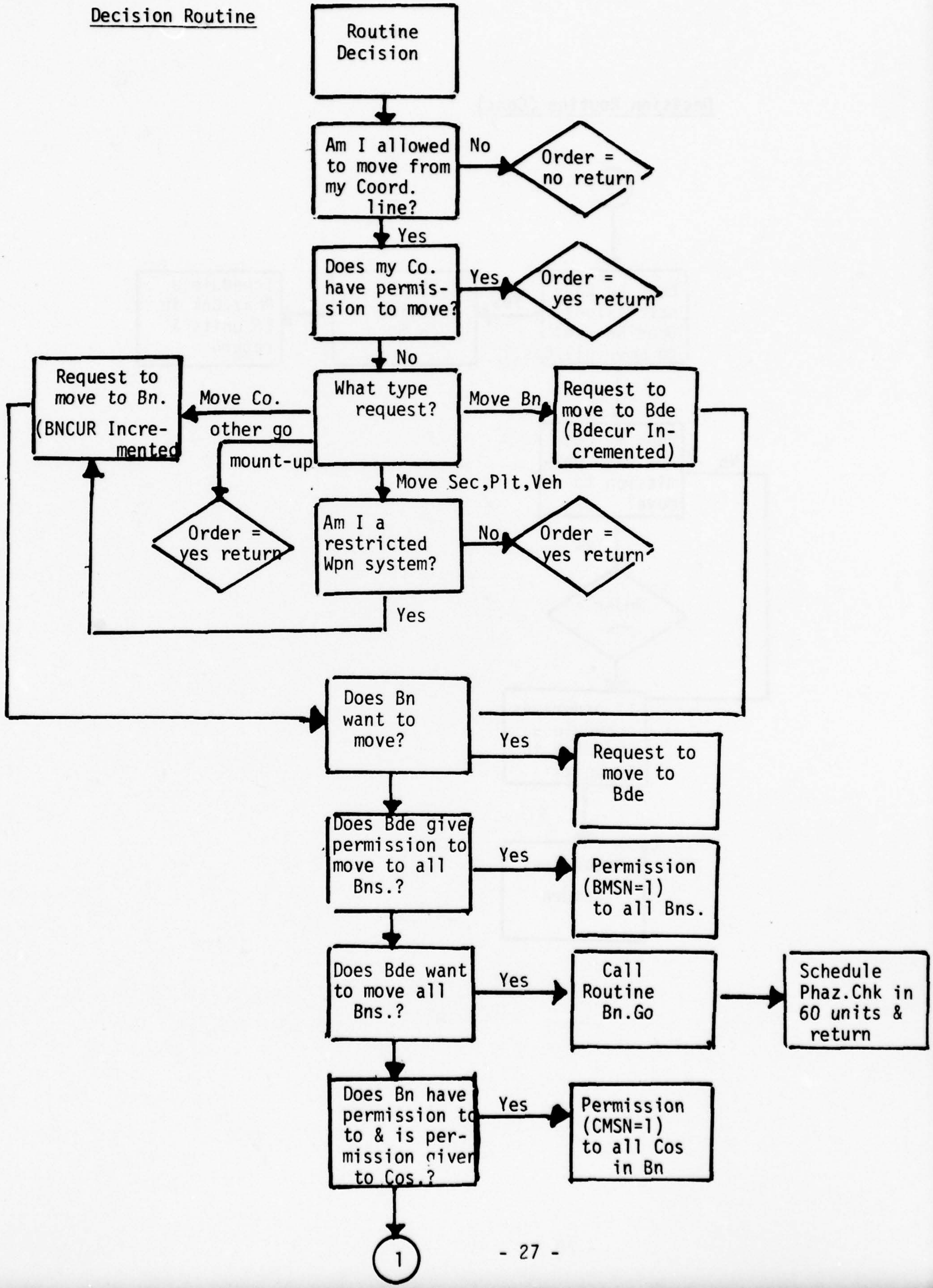
It will be useful at this point to trace thru the event flow to determine if a desired movement action is allowed to happen: the value of the variable order takes on the value 0/1 (0 = no, movement not allowed, 1 = yes movement is allowed). The term "restricted wpn system" refers to the $4^{\underline{th}}$ column of the Table array (1 = not restricted, 0 = restricted). For example, the tanks of a particular Company may be specified as a restricted weapon system by the user which in effect says this Company can't move any of its tanks without permission from higher headquarters. Perhaps the best way to understand the logic flow of requests to move is to trace through a flow chart of Routine Decision. Routine Decision is called when a unit, or vehicle wants to take some sort of movement action (e.g. move a platoon, move a company, move all IFV's, etc.). (See Table 1).

The Decision routine does essentially three things:

1) Determines if a unit is to request a move, and processes the request to the next higher headquarters.
2) Determines if a unit has permission to move.
3) Determines if a unit is ordered to move.

See Table 1 which sumarizes the request, permission, order logic for units.

Decision Routine



Decision Routine (Cont)

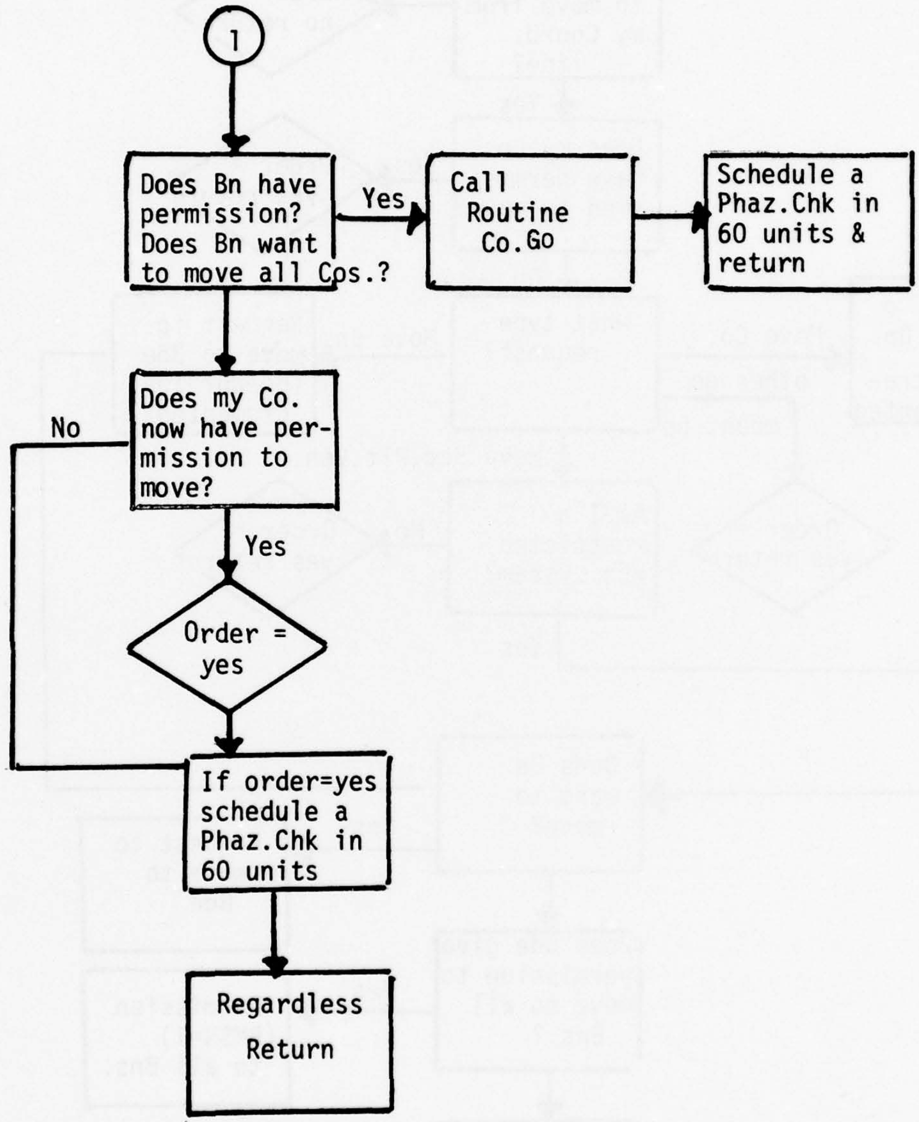


Table 1

Unit Request, Permission and Order Logic

	Company	Battalion	Brigade
Request	Co to Bn request- 1) Desire (from Action) to move Company. 2) Desire (from Action) to move a restricted weapon system	Bn to Bde request- 1) Desire (from Action) to move Battalion. 2) Lower bound on sum of Company weights reached	N/A
Permission to move	N/A	Permission granted for Co's to move: If Lower bound on sum of Company weights reached - and - Permission granted by Brigade	Permission granted for Bn's to move: If Lower bound on sum of Bn weights reached
Order to move	N/A	Order for Companys to move: If Upper bound on sum of Company weights reached - and - Permission granted by Brigade	Order for Bn's to move: If Upper bound on sum of Bn weights reached

V. - COUNTERATTACK LOGIC DOCUMENTATION -

1. General: Logic has been provided for the conduct of limited counterattacks within the model. Such counterattacks are pre-planned by the user and are executed if certain user specified conditions are met. The term counterattack used herein refers to limited offensive operations conducted within the context of the defense to inflict damage on an enemy unit whose offensive momentum has been reduced. It is important to make the distinction between counterattacks run on-line in the model and counter-offensive operations which might be conducted to reestablish previously occupied battle positions.

2. Assumptions:

a) Counterattacks are conducted only against enemy forces in hasty defense.

b) Counterattacks are conducted by company sized units.

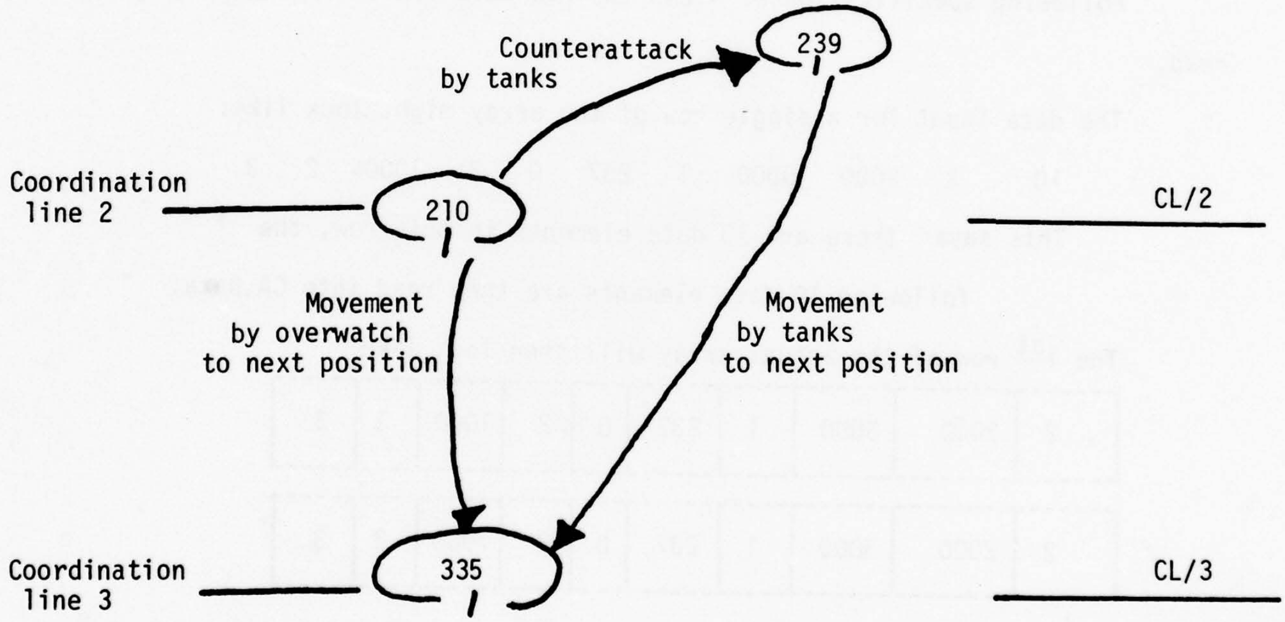
c) The assault is conducted only by tanks, ATGM firing vehicles remain in place to overwatch the assaulting forces.

d) The unit to be counterattacked must be within a certain distance of a user specified counterattack trigger point or the counterattack will not take place.

e) The company conducting the counterattack must be closer to the unit to be attacked by some user specified distance than is the closest enemy unit not being counterattacked. (See ROUTINE CHARGE).

f) The Blue/Red force ratio must be acceptable (user specified) or the counterattack will not occur.

3. <u>Technique</u>: The unit conducting the counterattack will attack from its present location to an area specified by the user. Upon arrival at the counter-attack area the counterattacking unit will resume normal defensive operations and move to its next specified defensive position when caused to redeploy by the movement decision/coordination logic of the simulation.



4. <u>Routines which impact on Counterattack Logic:</u>

   a. Routine Set.CA - reads data supplied by user into array CA.Data.

   b. Routine Ctr.Atk - determines if counterattack is to be conducted based on user supplied conditions.

   c. Routine Draw. Sabres - causes counterattacks to take place.

   d. Event Charge - calls routine Ctr.Atk.

5. <u>User Input</u>: User input for counterattack logic is read in the routine Set.CA. and read into the array CA.Data. The CA.Data array is a 2-dimensional ragged array, the dimensions of which must be specified by the user .

The user specifies the value of the local variable N.CAS which determines the number of rows to be created in the array and corresponds to the number of potential company level counterattacks to be conducted. If N.CAS=0 no counter-attacks data is read.

Following specification of N.CAS the row data for the CA.Data array is read.

The data input for a single row of the array might look like:

10 2 2000 3000 1 237 0 2 1000 2 3

This says there are 10 data elements in this row, the following 10 data elements are then read into CA.Data.

The 1st row of the actual array will then look like:

2	2000	3000	1	237	0	2	1000	2	3

2	2000	3000	1	237	0	2	1000	2	3

Coordination line from which counterattack is to be conducted.

	2000	3000	1	237	0	2	1000	2	3

X Coordinate, Y Coordinate of counterattack trigger point.

When a unit goes into hasty defense its actual location is compared with the X and Y of the trigger point. If the unit in hasty defense is not at the trigger point (within a certain tolerance) the counterattack

is not allowed to take place.

	1	237	0	2	1000	2	3

Number of the company to perform the counterattack.

	237	0	2	1000	2	3

Not used (<u>Always</u> enter 0)

Area to which assault will be conducted. Area.Start of each assaulting vehicle is its current area. Area.End of each assaulting vehicle is set to 237.

<u>Caution</u>: The route data specified by the user must include a route for the counterattacking company from its position on the coordination line to the area where the assault will terminate.

	2	1000	2	3

Acceptable Blue/Red force ratio

Friendly companies in a position to support the counterattacking company.

Routine Ctr.Atk determines the number of enemy elements in the unit to be assaulted, the number of friendly forces in the assaulting company, and the number of elements in each company which the user has specified as being in a position to support the counterattack. (<u>IF none enter 0</u>)

The total of friendly forces is divided by the number of elements in the enemy unit to be assaulted and this number is compared to the acceptable force ratio specified by the user.

If the actual force ratio is less than the acceptable force ratio, the counterattack is not allowed to occur.

	1000	

Distance factor.

It is essential that the decision to counterattack take into consideration the proximity of follow-on units or other echelons to the unit in hasty defense.

Routine Ctr.Atk calculates the distance from the assaulting company to the unit in hasty defense, and the distance of the closest follow-on unit to the unit in hasty defense.

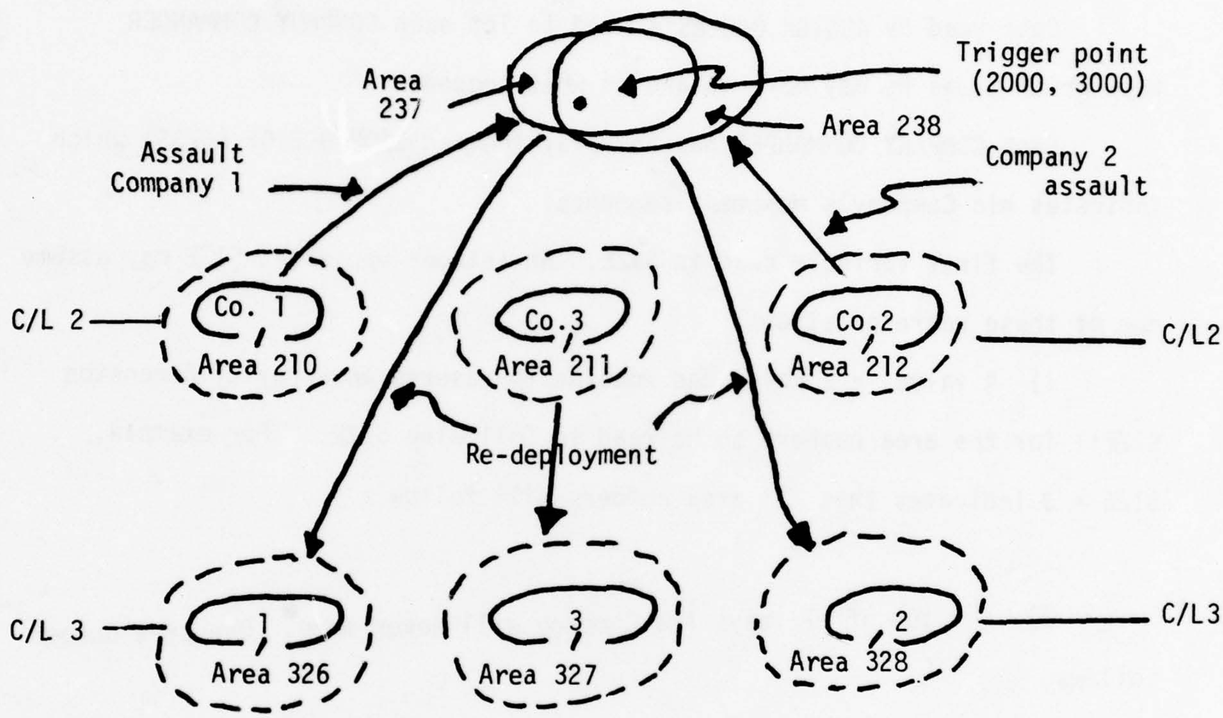
IF the assaulting company is not closer to the unit in hasty defense than the closest follow-on unit by at least this user specified distance factor, the counterattack is not allowed to occur.

6. Comments: The entire CA.Data array might look like

2	2000	3000	1	237	0	2	1000	2	3		
2	2000	3000	2	238	0	2	1000	1	3		
3	4000	8000	2	347	0	2	1500	1	3	4	
4	6000	8000	3	481	0	1	100	0			

Note that it is possible to conduct multiple company counterattacks by the clever use of the input data. For example, in rows 1 and 2 of the above array an enemy in hasty defense at trigger point 2000, 3000 would be attacked

by both companies 1 and 2. To illustrate how this might work:



Note also at row 4 of the above array, Company 3's assault from coordination line 4 is not supported by other companies, thus a 0 is entered in column 9 of the array.

VI. SEQUENCE OF MOVEMENT AREAS

Data read by ASSIGN.ORDERSis used to let each COMPANY.COMMANDER know which areas he may move to and in what sequence.

Each COMPANY.COMMANDER has an array (called SEQUENCE.OF.AREAS) which indicates his Company's movement sequence.

The first variable read is SIZE. An integer variable, SIZE may assume one of these representations:

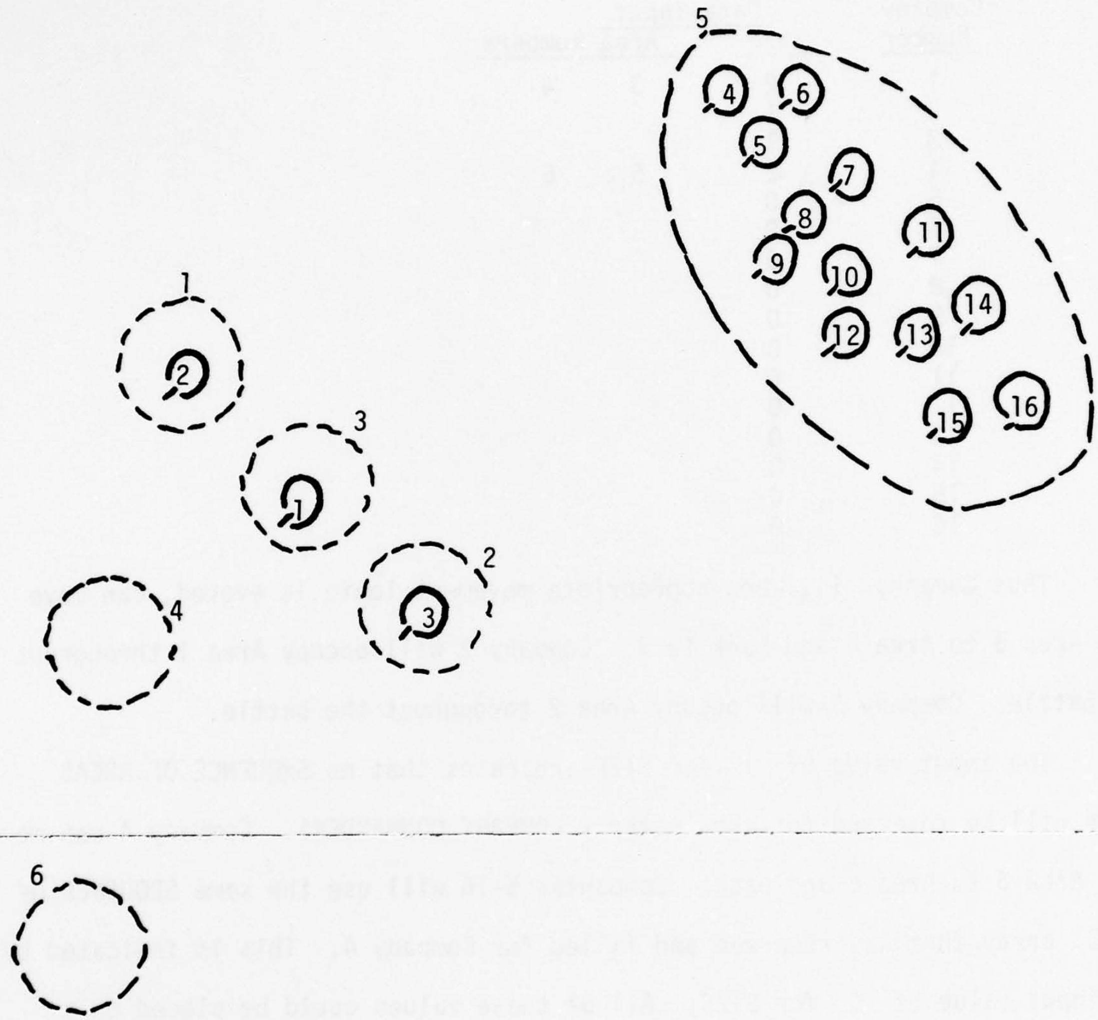
1) A value ≥ 2 keys the routine to reserve an array of dimension SIZE+1 for the area numbers to be read in following SIZE. For example, SIZE = 3 indicates that 3 area numbers will follow .

2) A value of 1 says the Company will never move. One area number follows.

3) A value of 0 says to use the most recently created SEQUENCE.OF.AREAS array.

A figure in conjunction with some sample input should make this clear.

Assume 16 Companies; 3 Defenders (1,2,3), 13 Attackers (4-16).



Dashed lines indicate "areas" for which positions have been provided by the user. These features are explained in detail in Volume 5, STAR Movement Model).

Each Company's area information as input to routine ASSIGN.ORDERS would appear on a data card as:

Company Number	Data Input Size	Area Numbers	
1	2	3	4
2	1		
3	1		
4	2	5	6
5	0		
6	0		
7	0		
8	0		
9	0		
10	0		
11	0		
12	0		
13	0		
14	0		
15	0		
16	0		

Thus Company 1, when appropriate movement logic is evoked, can move from Area 3 to Area 4 and back to 3. Company 2 will occupy Area 1 throughout the battle. Company 3 will occupy Area 2 throughout the battle.

The input value of 1 for SIZE indicates that no SEQUENCE.OF.AREAS array will be reserved for the latter 2 COMPANY.COMMANDERS. Company 4 can move from Area 5 to Area 6 and back. Companies 5-16 will use the same SEQUENCE OF AREAS array that was reserved and filled for Company 4. This is indicated by the input value of 0 for SIZE. All of these values could be placed on a single card.

Discussion of the actual routes between areas and the positioning of forces within areas is given in Volume 5, STAR Movement Model.

VII. Target Selection Tactics

STAR currently provides 14 target selection tactics. A tactic may be entered for each system/weapon combination represented in the POINT.HOLD array by entering the crewdrill number as described in the discussion of input requirements for routine DANGER.STATE.

The following discussion is keyed to these crewdrill numbers:

1. Attempt to acquire your platoon leader's target. Failing this, search your platoon to determine which of your targets are not being engaged by another platoon member. From this reduced set, engage your highest priority target. If all targets are engaged, engage your highest priority target using the alternate ammunition type that you specified in array POINT.HOLD.
2. Attempt to acquire your platoon leader's target. Failing this, search your platoon to determine which of your targets are not being engaged by another platoon member. From this reduced set, engage your highest priority target. If all targets are engaged, engage your highest priority target with the ammunition specified by the target selection array (the ARRAY for this system/weapon combination).
3. Attempt to acquire your platoon leader's target. Failing this, search your platoon to determine which of your targets are not being engaged by another platoon member. From this reduced set, engage your highest priority target. If all targets are engaged, do not engage any target.
4. Same as 1., except company is searched.

5. Same as 2., except company is searched.

6. Same as 3., except company is searched.

7. Attempt to acquire your platoon leader's target. Failing this, engage your highest priority target regardless of its engagement status.

8. - 14. These tactics are identical to 1. through 7. except that no attempt is made to acquire the platoon leader's target.

Thus, crewdrills 1. through 7. attempt a platoon leader handoff as the first choice for a target selection tactic. Crewdrills 8. through 14. move directly to an evaluation of each target in the selector's list of detected targets.

If a platoon handoff is not accomplished (because it failed or was not specified) then the following statements apply:

1. Crewdrills 1., 4., 8., and 11. are "missile savers" in that if all targets are being engaged, an alternate ammunition type (usually some sort of automatic weapon ammunition) will be specified to engage the threat.

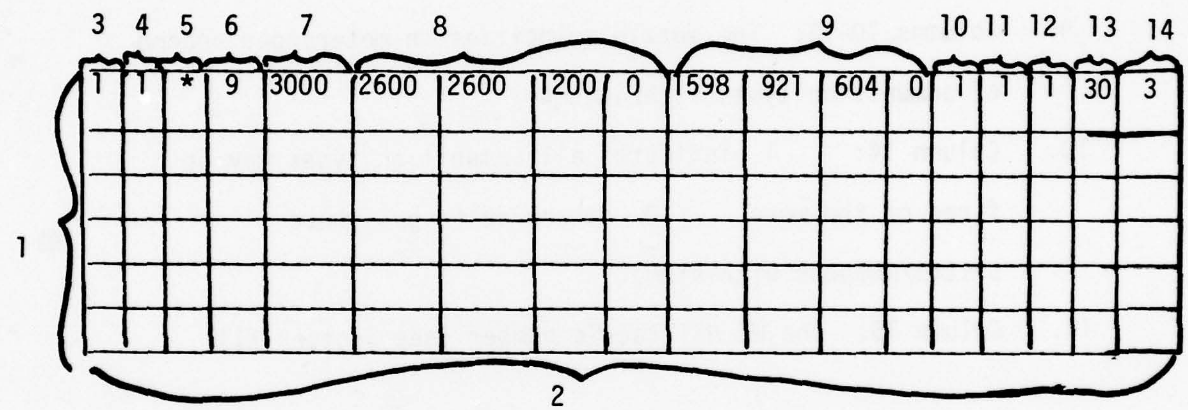
2. Crewdrills 2., 5., 9., and 12. will always result in the selection of a target if a target is available.

3. Crewdrills 3., 6., 10., and 13. will only allow selection of an unengaged target.

4. Crewdrills 7. and 14. will always select the highest priority target from a selector's list of detected targets, regardless of that target's current engagement status.

ROUTINE DANGER.STATE

Routine DANGER.STATE sets up array POINT.HOLD and array ARRAY, the System/Weapon target selection array. A diagram of POINT.HOLD and a specified ARRAY are shown below:



1. NUM.DS.ARRAYS: The number of rows of array POINT.HOLD and the number of ARRAYs to be created.
2. WD: The number of columns of array POINT.HOLD.
3. Column 1: SYSTYPE: The system type of the firer under consideration.
4. Column 2: WPNTYPE: The weapon type of the firer under consideration.
5. Column 3: The storage location of the pointer to the given system/ weapon type's target selection ARRAY.

6. Column 4: The target selection crew drill number for this system/weapon type (see previous discussion for available options).

7. Column 5: The acquisition range in meters for the system/weapon type.

8. Column 6-9: The maximum opening range in meters for ammunition type 1 through 4.

9. Columns 10-13: The muzzle velocities in meters per second of ammunition types 1 through 4.

10. Column 14: 1 indicates all ammunition types may be fired on the move 0 otherwise (e.g., those system/weapons with ATGM).

11. Column 15: The WE.HIT tactic number (see Section XI).

12. Column 16: The WE.MISS tactic number (see Section XI).

13. Column 17: Time in seconds to remain in full defilade after a WE.HIT/WE.MISS sequence.

14. Column 18: Alternate ammunition type for use in routine TACTICS.

An example of an ARRAY for a system type 2, weapon type 3 (an IFV for illustrative purposes) is shown below:

0-1000 meters	1	7	3	1	2	8	1	3	2	8	2	1
1000-2000 meters	1	7	11	1	2	8	12	1	2	8	13	3
>2000 meters	1	7	21	1	2	8	22	1				

The first row represents the 0-1000 m range band and has 3 blocks of 4 numbers. The first 4 numbers indicate that system type 1, weapon type 7 has a priority 3 using ammunition type 1 within this range band. Thus numbers are read in the following order:

System type of target

Weapon type of target

Priority of this target

Ammunition type to be used against this target.

Moreover, within system/weapon types, if more than one priority is to be assigned within a range band, the blocks of 4 numbers should be contiguous and the priorities increasing to the right.

If a firer may not engage targets in a given rangeband, a row of 4 zeros (0 0 0 0) should be input for that rangeband.

In the actual target selection process, each potential target in the element's detected list is evaluated as to its priority. Note from the previous table that targets in the 1000-2000 meter range band were assigned priorities 11, 12 and 13. Since the lowest numerical priority is selected (with the closest target selected for equal priority) range band 0-1000 will always have priority over band 1000-2000. If, for example, it is desired to have the 2000-3000 range band as highest priority, the user assigns the lowest numerical values of priority to target/ammo types in that range band.

Since POINT.HOLD and ARRAY are filled in one operation, some sample input for both arrays is shown (letters are keyed to the explanation below);

```
a  b
7  16

c  d  e    f       g
2  3  8    3500    3000 0 1500 0

              h
217  0        1132   0

i  j  k    k2    k3
0  3  3    20    3

l                    m
3     1 7 3 1    2 8 1 3    2 8 2 1

n                    o
3     7 11 1     2 8 12 1   2 8 13 3

p                 q
2     1 7 21 1   2 8 22 2
```

a. NUM.DS.ARRAYS

b. WD

c. SYSTYPE (Column 1 of POINT.HOLD)

d. WPNTYPE (Column 2 of POINT.HOLD)

e. Target selection crew drill number (Column 4 of POINT.HOLD)

f. Acquisition range (Column 5 of POINT.HOLD)

g. Opening range of ammo types (0 indicates ammo type is not available). (Columns 6-9 of POINT.HOLD).

h. Muzzle velocity of ammo types (0 indicates ammo type is not available). (Columns 10-13 of POINT.HOLD)

i. Fire on move capability (Column 14 of POINT.HOLD)

j. WE.HIT tactic number (Column 15 of POINT.HOLD)

k_1. WE.MISS tactic number (Column 16 of POINT.HOLD)

k_2. Defilade time

k_3. Alternate ammo for use in routine TACTICS

l. Number of 4 number blocks of target selection information to be entered in the 0-1000 meter rangeband.

m. The 0-1000 meter rangeband target selection information.

n. As in l, except for 1000-2000 meter rangeband.

o. As in m., except for 1000-2000 meter rangeband.

p. As in l, except for > 2000 meter range band.

q. As in m, except for > 2000 meter range band.

A block of data in the manner described by c. through q is to be input for each system/weapon under consideration.

VIII. XM-1 AMMUNITION REDISTRIBUTION

STAR is prepared to redistribute ammunition from storage compartments on the XM1 to the ready bustle. In general, ammunition is redistributed during a period of full defilade. Rounds are moved to the ready bustle and appropriate bookkeeping is accomplished. Moreover, vulnerability changes as ammunition is redistributed.

If movement to defilade is not played in a particular run, redistribution of ammunition is accomplished while moving.

If any row total is equal to 0, the reload time (e.g., CATIME, SITIME, S2TIME, etc.) is also input as 0.

Redistribution of ammunition is meaningful for XM1's defined as follows:

XM1/105mm	Sys.Type = 1	Wpn.Type = 1
XM1/120mm	Sys.Type = 1	Wpn.Type = 2

First, a brief review of the variables:

CASEAP, CASEHE: Number of APDS and HEAT rounds initially loaded on XM1 (CASEAP+CASEHE=40,43,49,52, or 55).

CAPDS, CHEAT: Number of APDS and HEAT rounds initially loaded in bustle ready compartment.

CDA, CDH: As above, loaded in compartment next to left front fuel tank and in swing basket

BA, BH: As above, loaded in hull compartment

S1A, S1H, S2A, S2H: As above, loaded in bustle semi-ready compartment.

Input restrictions for the 52, 49, 43, and 40 round XM1 (120mm cases and the 55 round XM1/105mm case as shown in the following tables:

Note: Any APDS-HEAT combinations that satisfy the row and column totals in the tables are permissable.

XM1/120mm 52 Round Case (Case 1)

APDS		HEAT		
CAPDS	12	CHEAT	3	15
CDA	8	CDH	4	12
BA	4	BH	2	6
S1A	9	S1H	6	$5 \leq S1A+S1H \leq 15$
S2A	2	S2H	2	S2A+S2H=19-(S1A+S1H)
CASEAP = 35		CASEHE = 52-CASEAP=17		52

XM1/120mm 49 Round Case (Case 2)

APDS		HEAT		
CAPDS	12	CHEAT	3	15
CDA	6	CDH	3	9
BA	4	BH	2	6
S1A	9	S1H	6	$5 \leq S1A+S1H \leq 15$
S2A	2	S2H	2	S2A+S2H=19-(S1A+S1H)
CASEAP = 33		CASEHE = 49-CASEAP=16		49

XM1/120mm 43 Round Case (Case 3)

APDS		HEAT		
CAPDS	12	CHEAT	3	15
CDA	2	CDH	1	3
BA	4	BH	2	6
S1A	9	S1H	6	$5 \leq S1A+S1H \leq 15$
S2A	2	S2H	2	S2A+S2H=19-(S1A+S1H)
CASEAP =29		CASEHE = 43-CASEAP=14		43

XM1/120mm 40 Round Case (Case 4)

APDS		HEAT			
CAPDS	12	CHEAT	3	15	
CDA	0	CDH	0	0	
BA	4	BH	2	6	
S1A	9	S1H	6		$5 \leq S1A+S1H \leq 15$
S2A	2	S2H	2		S2A+S2H=19-(S1A+S1H)
CASEAP = 27		CASEHE = 40-CASEAP=13		40	

XM1/105mm 55 Round Case

APDS		HEAT		
CAPDS	16	CHEAT	6	22
CDA	7	CDH	4	11
BA	0	BH	0	0
S1A	0	S1H	0	0
S2A	16	S2H	6	22
CASEAP = 37		CASEHE= 55-CASEAP=18		55

A sample card representing the numbers from the first table (XM1/120mm 52 Round Case) is shown here:

35 17 12 3 8 4 4 2 9 6 2 2

DEFTIME,CDTIME,BTIME,S1TIME,S2TIME

Depending on the selection of WE.HIT and WF.MISS tactics, some systems may be played as moving to a fully defiladed hide position following a specified firing sequence. If any system is to hide, then DEFTIME should be input as a positive number representing the number of seconds in the fully defiladed position.

XM1's are required to effect ammunition redistribution during the battle

(see previous discussion for specific redistribution configurations). If a row total from the sample tables discussed previously is > 0, then the associated time variable for distributing that number of rounds to the ready bustle should be input. If a row total is 0, the associated reload time should be input as 0 . An exception to this discussion occurs when DEF.TIME is set to 0, indicating no movement by any system to dull defilade. In this case all the time variables should be set to 0. Redistribution is still accomplished, but it is accomplished while the vehicle is in an exposed position (defilade numbers 2 through 5).

An example for the XM1/120mm 52 Round Case with vehicles using the hide tactic is shown here:

20.0 210.0 108.0 125.0 45.0

An example for the above situation with no hide tactic is shown here:

0,0 0.0 0.0 0.0 0.0

The definitions of these times are as follows:

CDTIME: total time to move rounds from the fuel compartment/basket to ready bustle (sec).

BTIME: total time to move rounds from the hull compartment to the ready bustle (sec).

S1TIME: total time to move first increment of rounds from semi-ready to ready bustle (sec).

S2TIME: second increment analog of S1TIME.

IX. Ammunition Resupply

Integer Variables:

PCF1, PCF2, PCF3, PCF4, PCF5, PCF6 - Supply vehicles may carry any of 6 ammunition types from the Ammunition Transfer Point (ATP) to the Ammo. Pile. However, vehicles are loaded with pallets of ammo which must be converted to the actual number of rounds when the ammo is delivered to the AMMO.PILE. This is accomplished by the Pallet Conversion Factors which are input by the user and read by routine PILE.SO.CREATE. For example, PCF1 might be used to convert pallets of tank APDS ammo to actual number of rounds. PCF1 would by input as an integer 30. The PCF_i, their values, and the ammunition types they correspond to are indicated below (compatible with current UPLOAD event):

PCF1:	30	(XM1 APDS 120mm)
PCF2:	30	(XM1 HEAT 120mm)
PCF3:	12	(TOW)
PCF4:	20	(DRAGON)
PCF5:	500	(25mm BUSHMASTER)
PCF6:	1500	(.50 Caliber machine gun)

OFFLOAD - The estimated time in seconds to place pallets of ammunition on the ground at the AMMO.PILE. Input in MAIN.

LOADUP - The estimated time in seconds to place pallets of ammunition on the supply vehicle at the ATP. Input in MAIN.

NUM.PILES - The number of AMMO.PILE temporary entities to be created. Note that an ATP is not an AMMO.PILE. Input in routine PILE.SO.CREATE. Dimensions rows of array PLACES.

COL - The number of columns of array PLACES. Input in routine PILE.SO.CREATE.

TOW1CASE - The number of TOW rounds initially allocated to Infantry Fighting Vehicles or Cavalery Fighting Vehicles. Input in routine PILE.SO.CREATE.

TOW2CASE - The number of TOW rounds initially allocated to Improved TOW Vehicles. Input in routine PILE.SO.CREATE.

DRAGON - The number of DRAGON rounds initially allocated to each DRAGON missile team.

Integer Arrays:

PLACES - A 2-dimensional integer array used to store information pertinent to a particular AMMO.PILE. This array is used by supply vehicles to determine their movement areas, the pointer variables to specific AMMO.PILES, the ground force area associated with the AMMO.PILE, and the predetermined pallet loading plan for supply vehicles servicing a particular AMMO.PILE. This array is dimensioned NUM.PILES by COL. An example of PLACES is shown in Figure 1. It is suggested that the reader peruse this example now and then refer to it again following the discussion of the temporary entity AMMO.PILE.

N.SUPPLY.OFFICERS - Number of Supply Officers

Permanent Entities and Attributes

SUPPLY.OFFICER - This entity is used to define the limit of responsibility in the PLACES array through the use of the attribute MAX.POS.POINT. MAX.POS.POINT is set to the index of the last row in the PLACES array for which the particular SUPPLY.OFFICER is responsible. Additionally, each SUPPLY.OFFICER owns a set, STOCKS, which contains the AMMO.PILEs belonging to that SUPPLY.OFFICER.

Temporary Entities and Attributes

AMMO.PILE - This entity represents an area on the ground to which ammunition is delivered by supply vehicles and from which ammunition is

loaded on blue ground force vehicles. Each AMMO.PILE belongs to a set, STOCKS, described previously. Attributes of each AMMO.PILE are described below:

D1 through D6 - the number of rounds of ammunition in each of 6 user defined types to be delivered to the AMMO.PILE.

S1 through S6 - the number of rounds of ammunition in each of 6 types actually delivered to the AMMO.PILE.

OH1 through OH6 - the number of rounds of ammunition currently on-hand (available) at the AMMO.PILE . These numbers represent the difference between the amount sent and the amount uploaded by a ground force "customer."

LOCATION - The area number* of the AMMO.PILE.

GFAREA - The area number* of the ground force unit which is being supplied by AMMO.PILE at LOCATION.

ATP.AREA - the area number* of the ammunition transfer point servicing a particular AMMO.PILE.

STATUS - Assumes any of 3 values depending on the fill condition of the AMMO.PILE:

- 0 ≡ No ammunition delivered
- 1 ≡ Some ammunition received
- 2 ≡ AMMO.PILE has been filled with desired amounts of ammunition. An AMMO.PILE is also assigned a status of 2 if it is abandoned (e.g., all ground force units have departed the area being serviced by the AMMO.PILE).

MY.SO - The index number of the SUPPLY.OFFICER responsible for this AMMO.PILE. Used to file the AMMO.PILE in the appropriate STOCKS set.

* A detailed discussion of area numbers is given in Volume 5, STAR Movement Model.

Event Notices

UPLOAD - An event used to "top-off" a ground force element with the required ammunition, if available. This action occurs when the ground force element reaches its new location in an alternate position. An amount of time is assessed during which the element is placed in full defilade. If ammunition is not available when the element reaches his position, no UPLOAD will occur while he is at that position even though ammunition may later be delivered to the area.

Requires the pointer variable of the ground force element as an argument scheduled by routine SET.MOVE.AREAS.

OFF.LOAD - An event used to remove pallets of ammunition from a supply vehicle at the desired AMMO.PILE. Pallets are then converted to rounds by type and added to the amount sent (S_i) and the amount on hand (OH_i). OFF.LOAD occurs when the supply vehicle reaches the AMMO.PILE LOCATION and takes OFFLOAD units of time to accomplish. This event is scheduled by routine WHERE.AM.I and requires the pointer variable of the supply vehicle as an argument.

MOVE.OUT - An event used to initiate movement from ATPs to AMMO.PILEs and the converse. Requires the pointer variable to the supply vehicle as an input. Scheduled by events LOAD.PLAN and OFF.LOAD, and by routine WHERE.AM.I.

LOAD.PLAN - An event used to simulate the loading of pallets of ammunition on supply vehicles at the ATP. An amount of time equal to LOADUP is assessed. Pallet load plans by system/weapon type of supply vehicle with respect to a particular AMMO.PILE are stored in array PLACES. Scheduled initially by routine SET.AREAS and subsequently by routine WHERE.AM.I

Assume 2 Supply Officers, each with 2 Supply Vehicles (GOER: Sys.Type = 7, Wpn.Type = 1; ARSV: Sys.Type = 7, Wpn.Type = 2). Each Supply Officer is responsible for 2 ATPs. Each ATP services 2 AMMO.PILES.

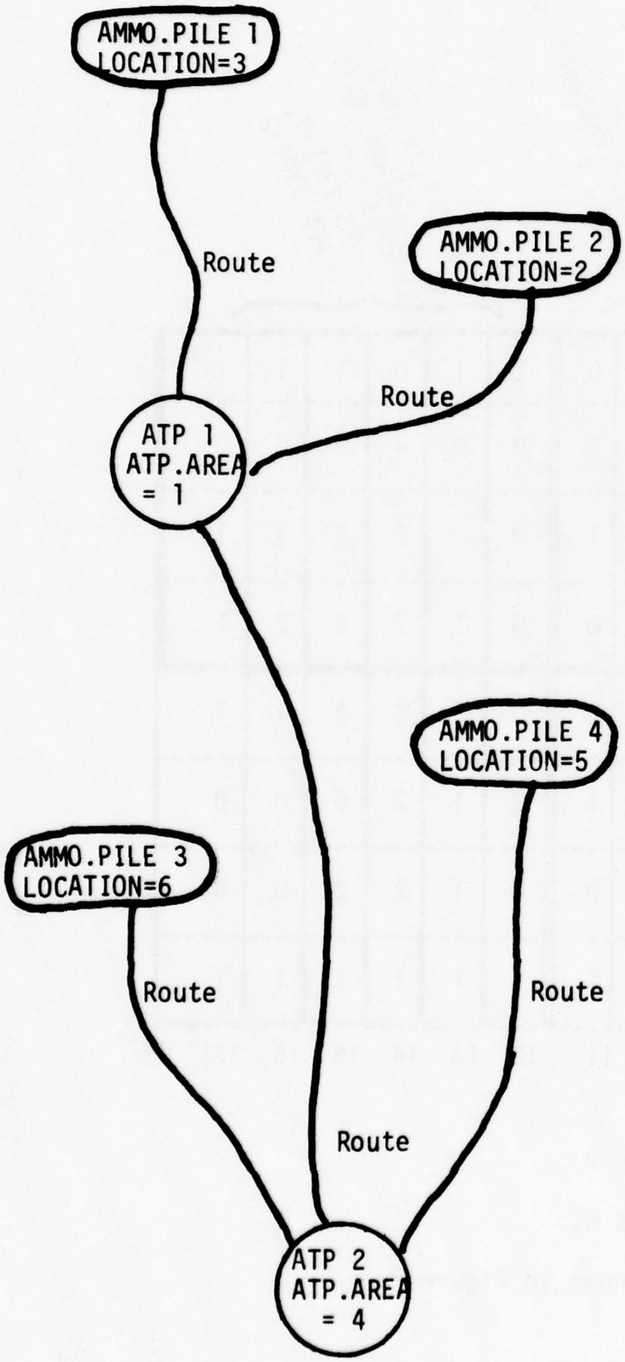
I \ J	1 Location	2 GFAREA	3 Pointer Variable to AMMO.PILE	4 MY.SO	5 ATP.AREA	6 LOAD PLAN INFO Pallets of AMMO 1-6 to be transported by SYS.TYPE=7, WPN.TYPE=1 to AMMO.PILE represented in Row I	7	8	9	10	11	12 LOAD PLAN INFO As in Columns 6-11 except for SYS.TYPE=7, WPN.TYPE=2	13	14	15	16	17
1	3	3	*	1	1	3	0	2	0	0	0	1	1	0	1	1	1
2	2	2	*	1	1	5	0	0	1	0	0	0	0	4	0	0	0
3	6	6	*	1	4	0	2	2	0	0	1	0	0	2	3	1	1
4	5	5	*	1	4	4	1	1	0	0	0	0	1	2	0	2	1
5	9	9	*	2	7	2	1	2	0	1	0	0	2	0	2	1	1
6	8	8	*	2	7	1	0	2	2	0	1	2	1	2	0	0	0
7	12	12	*	2	10	0	2	4	0	0	0	0	1	2	2	0	0
8	11	11	*	2	10	3	1	0	1	1	1	1	1	1	1	1	1

MAX.POS.POINT for <u>SUPPLY.OFFICER</u> number 1 is 4;

MAX.POS.POINT for <u>SUPPLY.OFFICER</u> number 2 is 8.

A schematic of the ATPs and AMMO.PILEs is shown in Figure 2.

Figure 1.
PLACES ARRAY EXAMPLE



The resupply plan established by SUPPLY.OFFICER number 1 calls for fill of AMMO.PILEs 1 and 3 in order, followed by a movement to ATP2 and fill of AMMO.PILEs 3 and 4 in order. The SUPPLY.OFFICER has available 1 GOER and 1 ARSV which will transport pallets as indicated in the places array. Note that if a D_i is > 0, then the associated load plan should call for movement of a positive amount of ammo type i. Otherwise, the status of the AMMO.PILE will only change to 2 when the AMMO.PILE is abandoned by the ground forces. LOCATION and ATP.AREA are placed on the Supply Vehicle as AREA.START and AREA.END as required by routine MOVE.

Figure 2.

SUPPLY PLAN EXAMPLE

Figure 3

Sample Ammunition Weight/Volume Calculations

	L		W		H			
TOW	56.5"	x	11.5"	x	11.5"	87 lbs.	4.4 ft^3	1 rd
	58.3	x	48	x	42	1100 lbs.	68 ft^3	12 rds
DRAGON	47.5	x	16	x	16	67 lbs.	7 ft^3	1 rd
	80	x	47.5	x	70	1400 lbs.	154 ft^3	20 rds
105 T	45.8	x	14.3	x	8.8	145 lbs	3.3 ft^3	2 rds (box)
HEAT	42	x	45	x	50	2255 lbs (2506)	54 ft^3	30 rds (15 boxes)
105 T	39.8	x	14.1	x	8.7	126 lbs	2.8 ft^3	2 rds
APDS	42	x	39	x	49	1970 lbs (2189)	46 ft^3	30 rds/pallet

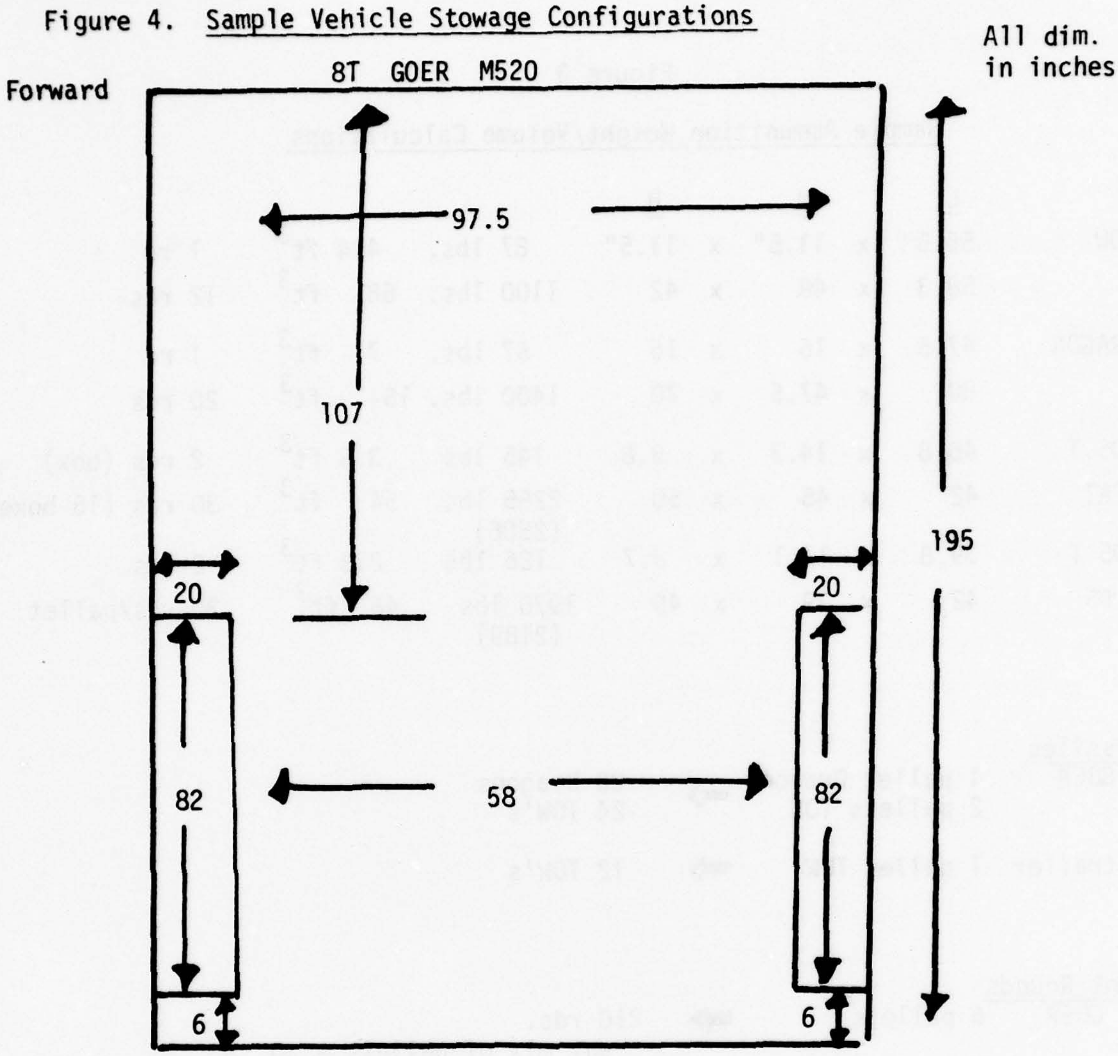
Missiles

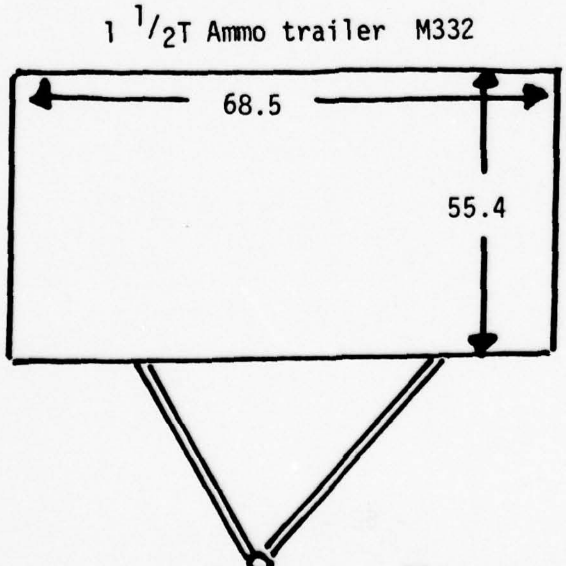
GOER	1 pallet Dragon 2 pallets TOW	➡	20 Dragons 24 TOW's
trailer	1 pallet TOW	➡	12 TOW's

Tank Rounds

GOER	6 pallets	➡	210 rds.
			any mix of pallets = $\frac{30}{7}$
trailer	1 pallet		

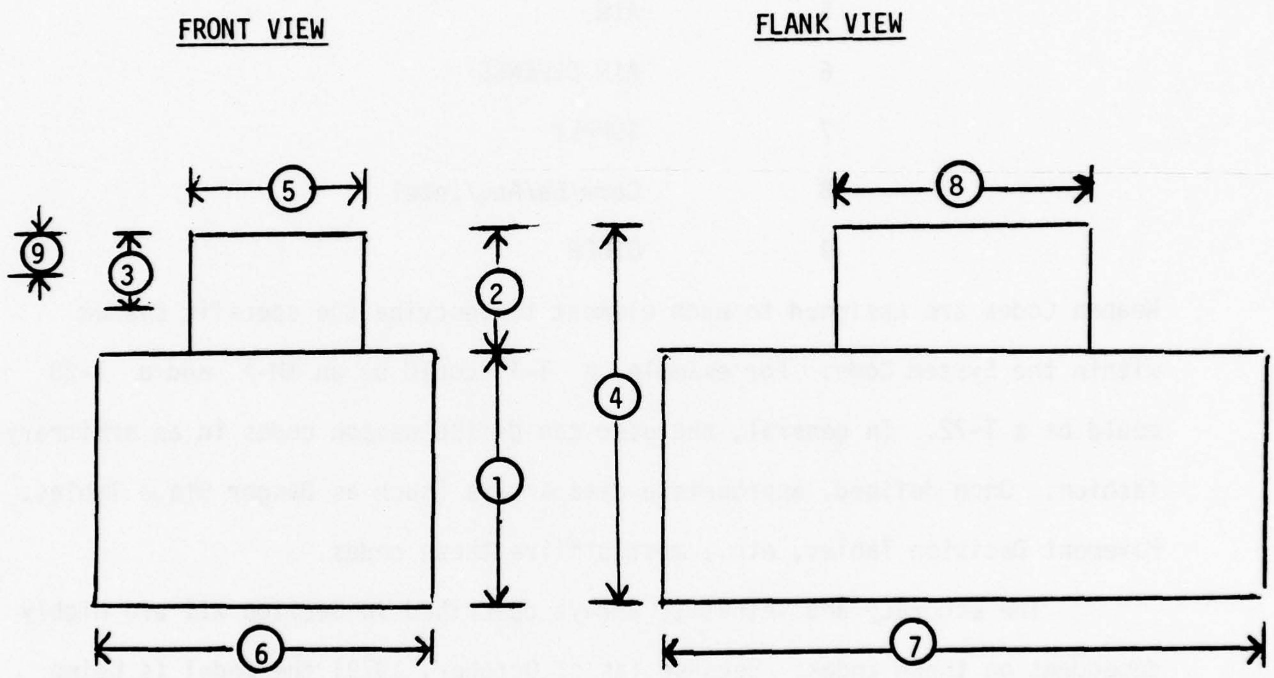
Figure 4. Sample Vehicle Stowage Configurations





X. TARGET DIMENSIONS and VARIOUS CODES

1. TARGET DIMENSIONS - Elements in STAR are represented as two rectangular solids described below. A diagram with dimensions numbered is provided, followed by a definition of those dimensions not obvious from the figure.



DEFINITION of Dimensions

3 - Exposed height of element in firing defilade

9 - Exposed height of element in turret defilade (detection mode)

10 - (not shown) is dimension 2 divided by dimension 4 which is the percent of total vehicle height above hull defilade line.

2. Elements in STAR are assigned a System Code which represents the general class of the element. These codes are as follows:

1	TANKS
2	MOUNTED INFANTRY
3	DISMOUNTED INFANTRY
4	ARTILLERY
5	AIR
6	AIR DEFENSE
7	SUPPLY
8	Comm/EW/Acq/Intel
9	OTHER

Weapon Codes are assigned to each element to describe the specific system within the System Code. For example, a 1-1 could be an XM-1 and a 1-28 could be a T-72. In general, the user can define weapon codes in an arbitrary fashion. Once defined, appropriate data arrays (such as Danger State Tables, Movement Decision Tables, etc.) must utilize these codes.

The accuracy and lethality arrays described in Section XII are highly dependent on these codes. Because (as of October, 1979) the model is being executed with modified lethality arrays from early 1979, the following weapon codes must be used to execute the current model.

Subsequent modifications to STAR will remove this restriction.

WEAPON CODES	ELEMENT TYPE	WEAPON CODES	ELEMENT TYPE
1	XM1/105mm	5	DIVAD Gun
2	XM1/120mm	6	DRAGON Msl Team
3	IFV	7	T72
4	ITV	8	BMP
		9	ZSU 23-4

3. <u>AMMUNITION CODES</u> - Each System/Weapon Code may be assigned up to four Ammunition Codes. These Codes (similar to Weapon Codes) are user defined, but all appropriate data arrays must be constructed after defining these Codes.

In order to execute the model (ground only) with current accuracy/lethality arrays, the following Ammo Codes are used:

1 - APDS for Tank Main Gun/TOW for IFV,ITV, ATGM for BMP

2 - HEAT for Tank Main Gun/DRAGON for dismounted crews

3 - Automatic Weapon for Tanks/BUSHMASTER for IFV/73mm for BMP

4 - Automatic Weapon

4. <u>OTHER CODES</u> - Although not required as input data, it is desirable that the user understand the following coded variables:

DEFNUM - describes the current postion/activity of an element

1 - Full defilade

2 - Turret defilade

3 - Firing defilade

4 - Stopping to fire

5 - Moving

6 - Reached final area in movement

XI. Defender Movement to Full Difilade Tactics

1. The integer values of WH.1, WH.2, WH.3, WH.4, WH.5, WM.1, WM.2 and WM.3 are used by Routines WE.HIT and WE.MISS to determine the number of rounds that may be fired prior to activating a user specified tactic. Variables preceded by WH are used by WE.HIT immediately following a catastrophic kill from a firing defender element and those preceded by WM are used by WE.MISS immediately following any other result. Throughout this discussion Hit should be understood to imply catastrophic Kill.

Three tactics are currently represented.

TACTIC 1: Specifies the shot sequence (using WH.1, WH.2, WH.3, WH.4, WM.1, WM.2) which will be followed by a move to full defilade by the defender element.

The logic sequence is as follows:

A. Following a hit

Go to full defilade if (since last defilade) (No. hits = WH.1) or (No. misses = WH.2) or (No. misses = WH.3 and No. hits = WH.4)

B. Following a miss

Go to full defilade if (since last defilade) (No. misses = WM.1) or (No. hits = WM.2)

TACTIC 2: A Target Selection event follows each shot, which implies the defender element will never go to full defilade.

TACTIC 3: Specifies the total number of rounds fired since last defilade which will cause defender element to go to full defilade.

The logic sequence is as follows:

Total number of shots $\geq$ WH.5 following a hit.

Total number of shots $\geq$ WM.3 following a miss.

EXAMPLE: Assume values are specified for WH.1,...,WH.5, WM.1, ...,WM.3 as follows:

2 2 1 1 2 2 2 2

As shown. WH.1 through WH.4, when used with WM.1 and WM.2, specify the following shot sequence followed by a movement to full defilade (HIDE). (WE.HIT and WE.MISS tactic 1):

HIT SELECT HIT HIDE

HIT SELECT MISS HIT HIDE

HIT SELECT MISS MISS HIDE

MISS HIT HIDE

MISS MISS HIDE

WH.5 and WM.3 specify the following shot sequence for systems that can move to full defilade (WE.HIT and WE.MISS tactic 3):

MISS SELECT MISS HIDE

MISS SELECT HIT HIDE

HIT SELECT MISS HIDE

HIT SELECT HIT HIDE

XII. Accuracy and Lethality Data

The routines which reserve the arrays for and read the data for accuracy and lethality data are RES2, RES3, RES4 and RES5. Twenty one arrays ranging from two to six dimensions are used to store the data. The arrays, their use, their dimensionality and the meaning of the dimensions are below.

ACCURACY ARRAYS

ADDON: Add on biases (in mils) in deflection and elevation for moving firers

3 dimensional Reserved as 2 by 2 by 10

1st Dimension: Type Vehicle
- 1 Tanks
- 2 BMP firing 73mm

2nd Dimension: Specific Error
- 1 Deflection Add on
- 2 Elevation Add on

3rd Dimension: Speed of the firer in 4mph increments.

0-2 = 1, 2-6 = 2, etc.

ACCBM: Accuracy Data for the BMP firing the 73mm cannon at a stationary target (for moving targets see BM.MOV)

3 dimensional. Reserved as 3 by 7 by 2

1st Dimension: Sensing of Last round fired at this target

1 First Round

2 Hit

3 Sensed Miss

2nd Dimension: Range to target (meters)

1 0-299
2 300- 499
3 500 699
4 700 899
5 900 1099
6 1100 1299
7 1300 1600

3rd Dimension:

1 Deflection sigma

2 Elevation sigma

ACCHT: Accuracy Data for Tanks firing Heat round at stationary target. See HT.MOV for moving targets

Four dimensional. Reserved as 2 by 4 by 7 by 2.

1st Dimension: Type of Tank

1 US Tank

2 OPFORCE Tank

2nd Dimension: Sensing of Last Round fired at this target

1 First Round

2 Hit

3 Lost Miss

4 Sensed Miss

3rd Dimension: Range to Target (meters)

1 0 - 349

2 350 - 749 and so on in 500 m increments

7 2250 - 2750

4th Dimension: Specific error

1 Deflection Sigma

2 Elevation Sigma

ACCMSL: Accuracy data for antitank guided missiles fired at moving or stationary targets.

Four dimensional. Reserved as 2 by 2 by 7 by 4

1st Dimension: Long or short Range ATGM

1 Long Range TOW, SAGGER

2 Dragon

2nd Dimension: Moving or Stationary Target

1 Stationary

2 Moving

3rd Dimension: Range to Target

	Long Range		Short Range	
1	0	374	0	149
2	375	749	150	249
3	750	1249	250	349
4	1250	1749	350	449
5	1750	2249	450	649
6	2250	2749	650	849
7	2749	3250	850	1049

4th Dimension: Specific Error

1 Deflection Bias
2 Elevation Bias
3 Deflection Sigma
4 Elevation Sigma

ACCKE: Accuracy data for tanks firing kinetic energy rounds at stationary targets. For moving targets see KE.MOV

Four Dimensional. Reserved as 2 by 3 by 7 by 2

1st Dimension: Type of Tank

1 US Tank

2 OPFORCE Tank

2nd Dimension: Sensing of Last Round fired at this target

1 First Round

2 Hit

3 Lost Miss

3rd Dimension: Range to target

See ACCHT

4th Dimension: Specific Error

1 Deflection Sigma

2 Elevation Sigma

BM.MOV: Accuracy Data for BMP firing 73mm HEAT Round at moving targets.

Three Dimensional. Reserved as 5 by 7 by 3

1st Dimension: Speed of the Target (Kph)

1	0 - 5
2	5 - 15
3	15 - 25
4	25 - 35
5	35 +

2nd Dimension: Range to the target (meters)

See ACCBM

3rd Dimension : Specific Error

1	Deflection Bias
2	Deflection Sigma
3	Elevation Sigma

BUSHBMP: Accuracy Data for an IFV engaging a BMP using the BUSHMASTER Weapon System

Five Dimensional. Reserved as 2 by 2 by 2 by 4 by 8

1st Dimension: Motion of shooter

1	Stationary
2	Moving

2nd Dimension: Motion of target

1	Stationary
2	Moving

3rd Dimension: Target disposure

1	Hull defilade
2	Fully exposed

4th Dimension: Type of Data

1	Mobility Damage Function
2	Firepower Damage Function
3	Probability of Catastrophic Kill
4	Probability of Hit

5th Dimension: Range to target (meters)

1	0 - 300 meters
2	300 - 600
3-8	400 meter range bands

DGNV: Probability of hitting a Dragon ATGM Team with secondary weapon system. A Dragon Team hit more than once by a single burst is assumed catastrophically killed, as in a Team hit by a Tank main gun round or ATGM.

Two Dimensional. Reserved as 2 x 8

1st Dimension: Firing System

1 Tank Machine Gun
2 BMP Heat Round

2nd Dimension: Range to Target

See BUSHBMP

HT.MOV: Accuracy Data for a tank firing a Heat round at a moving target.

Four Dimensional. Reserved as 2 by 5 by 7 by 3

1st Dimension: Type of Tank

1 US Tank
2 OPFORCE Tank

2nd Dimension: Speed of Target. (KPH)

See BM.MOV

3rd Dimension: Range to Target (Meters)

See ACCHT

4th Dimension: Type of Data

1 Deflection Bias
2 Deflection Sigma
3 Elevation Sigma

KE.MOV: Accuracy Data for a tank firing a kinetic energy round at a moving target.

Four Dimensional. Reserved as 2 by 5 by 7 by 3

1st Dimension: Type of Tank

1 US Tank
2 OPFORCE Tank

2nd Dimension: Speed of the target

See BM.MOVE

3rd Dimension: Range to the target

See ACCHT

4th Dimension: Type of Data

See HT.MOV

SOVMG: Data for OPFORCE tanks firing machine guns

Four Dimensional. Reserved as 2 x 3 by 4 by 8

1 US Tank
2 Other US vehicle

2nd Dimension: Target Disposure

1 Hull Defilade
2 Fully Exposed
3 Moving

3rd Dimension: Type of Data

1 Mobility Damage function
2 Firepower Damage function
3 Probability of Kill
4 Probability of Hit

4th Dimension: Range to target (meters)

See BUSHBMP

USMG: Data for US Tanks firing machine guns

Three Dimensional. Reserved as 3 by 4 by 8

1st Dimension: Target Disposure

1 Hull Defilade
2 Fully exposed
3 Moving

2nd Dimension: Type of Data

See SOVMG

3rd Dimension: Range to target

See BUSHBMP

LETHALITY ARRAYS

MINLETH: Data for Lethality of US mines against OPFORCE vehicles

1st Dimension: Type of Mine

2nd Dimension: Type of OPFORCE Vehicle

1 Tank
2 Other

3rd Dimension: Lethality

1 Mobility Damage Function
2 Firepower Damage Function
3 Probability of Catastrophic Kill

All other Lethality Arrays are name coded and fall into one of two groups. Data for Rounds whose lethality is range sensitive are stored in 6 dimensional arrays. For rounds whose lethality is not a function of range, the data are stored in 5 dimensional arrays. All array names are of the form LE**. The asterisks are replaced by numbers. The first number represents the system type of the vehicle, the second represents the type of munition.

Codes are as follows:

1st Number	1	US Tank
	3	Tow firing vehicle
	6	Dragon firer
	7	OPFORCE Tank
	8	BMP
2nd Number	1	Kinetic Energy or HEAT Round
	2	HEAT Round
	3	HEAT Round

The second number corresponds to the ammunition types (in order) carried by the vehicle. All lethality arrays share a common general structure.

5 Dimensional Arrays — Reserved as number of opposing systems by 2 by 10 by 3 by 7

1st Dimension:

Target Type	US Target	OP Force Target
1	Tank	Tank
2	Tank	BMP
3	IFV	ZSU
4	ITV	
5	DIVAD	

2nd Dimension: Target Disposure

1 Hull defilade
2 Fully Exposed

3rd Dimension: Dispersion Class in feet (1 to 10)

4th Dimension: Type of Data

1 Mobility Damage Function
2 Firepower Damage Function
3 Probability of Catastrophic Kill

5th Dimension: Aspect Angle in 30^{o} increments from 0^{o} to 180^{o}.

Six Dimensional Arrays. Reserved as number of opposing systems by 7 by 2 by 10 by 3 by 7.

1st Dimension: Same as 5 dimensional arrays.

2nd Dimension: Range to target (meters)

1 0 - 375
2 375 - 749
3-7 749 to 3000 in 500 meter increments

3rd thru 6th Dimension: Identical to 2nd thru 5th Dimensions of 5 dimensional arrays.

The 5 dimensional lethality arrays are

LE31	TOW
LE61	DRAGON
LE12	Tank Heat (US)
LE72	Tank Heat (OPFORCE)
LE81	BMP (Missile)
LE83	BMP (73mm Heat)

The 6 dimensional lethality arrays are

LE11	Kinetic Energy Round, U.S.
LE71	Kinetic Energy Round, OPFORCE

All the accuracy and lethality arrays are reserved by RES 2. RES 3 reads the LE** arrays in the following order. LE11, LE12, LE31, LE61, LE71, LE72, LE81, LE83. The data are read from logical unit 2 which must be established in the JCL for the run. The code which reads the data is a series of nested DO LOOPS. The data set should be stored on a series of records, each in the following format:

```
*  *  *  *  *  xxx xxx xxx xxx xxx xxx xxx
```

Where the asterisks are sort fields which will not be read, and the xxx's are the data (numbers 0 to 100) for a given set of parameters and seven aspect angles. 3600 such records are read. The format of the read is

```
FOR I = 1 to 2   FOR J = 1 to 7   FOR K = 1 to 12
FOR L = 1 to 10  FOR M = 1 to 3 DO
SKIP 5 Fields FOR N = 1 to 7 DO
Read LE11 (I,J,K,L,M,N)
LOOP LOOP
```

As the code is executed, the array indices are cycled in reverse order, and the data set must be stored accordingly. Note that the example is for a 6 dimensional array. For a 5 dimensional array the second index, (J = 1 to 7) which accounts for range, is omitted.

Routine RES4 reads the data for accuracy of principal weapon systems (Tank Main Gun, ATGM) using logical unit 3 for input. The arrays read are ACCKE, KE.MOV, ACCHT, HT.MOV, ACCMSL, ACCBM and BM.MOV in that order. Like the LE** data the accuracy data must be stored on a series of records. The format of the records is not as constant as for the LE** arrays due to the varying size of the arrays. The formats for each array are shown below, with * indicating fields which are not read, xxx indicating fields which are read. All arrays are real and data should normally contain decimal points.

```
ACCKE   * * * *             xxxx  xxxx
    42 records (REAL)

KE.MOV  * *  xxxx *         xxxx  xxxx
    70 records (REAL)

ACCHT   * * * *             xxxx  xxxx
    56 records (REAL)

HT.MOV  * *  xxxx     *     xxxx  xxxx
    70 records (REAL)

ACCMSL  * *  xxxx  xxxx     xxxx  xxxx
    28 Records (REAL)

BM.MOV  * *  xxxx     *     xxxx  xxxx
    70 Records (REAL)
```

As with the lethality arrays, the indices are cycled in reverse order. The code to read KE.MOV is:

```
FOR I = 1 to 2  FOR J = 1 to 5  FOR K = 1 to 7 DO
SKIP 2 FIELDS READ KE.MOV (I,J,K,1)
SKIP 1 FIELD FOR L = 2 to 3 READ KE.MOV (I,J,K,L)
LOOP
```

Several other lethality and accuracy arrays are read by Routine RES5. They are ADDON, DGNV, MINLETH, SOVMG, and BUSHBMP, and are read from logical unit 8. Again the data must be stored on a series of records of varying length, however, for these arrays there are no sort or dummy fields.

```
ADDON  (REAL)  4 records
    xxx  xxx  xxx  xxx  xxx  xxx  xxx  xxx  xxx  xxx

DGNV (INTEGER)  2 records
    xxx  xxx  xxx  xxx

MINLETH (INTEGER)  6 records
    xxx  xxx  xxx

SOVMG (INTEGER) 24 records
    xxx  xxx  xxx  xxx  xxx  xxx  xxx  xxx

BUSHBMP  (INTEGER)  32 records
    xxx  xxx  xxx  xxx  xxx  xxx  xxx  xxx
```

As for other arrays, the indices are cycled in reverse order.

It should be noted that in the interests of reducing the amount of memory space used to store these arrays, the integer arrays are packed, that is more than one value is stored in each word. Because of this, the following conventions must be strictly observed for integer arrays.

ALL LE** arrays and MINLETH numbers 0 to 100

a number GT 28-1 will write into the next cell.

DGNV, USMG, SOVMG, BUSHBMP Numbers 0 to 1000

Care must be taken to read numbers

Less than or equal to 2^{16}.

The structure of these arrays, the systems represented and the size of the numbers are integral parts of Routines COMPUTE and GEOM. If new systems which cannot be represented by one of the modeled systems are to be played, significant recoding of these routines, as well as of the RES* routines will be required.

XIII. SUPPRESSION MODULE

1. Introduction:

The play of suppression in STAR is designed to represent the effects of direct and indirect fire on delaying element functions. The play of suppression is parametric and the effects can be altered by the input parameters supplied by the user.

2. Assumptions:

a. The suppressive effect of a round is a decaying phenomenan, and can be represented as a time delay in the performance of element functions.

b. The suppressive effect of indirect fire is a function of the proximity of the impact to the target, and whether or not the round's impact is observed by the target.

c. Different rounds have different suppressive effects and can be represented by different round "weights".

d. The suppressive effect of direct fire rounds occurs if the round lands short, or if the round hits the target. Rounds which miss over a target are assumed to be unobserved and have no suppressive effect.

e. The susceptability to suppression of a particular weapon system can be represented by a parameter, and all similar weapon systems in the simulation have a common parameter

e.g. all XM1's have a $\lambda = 1$
all BMP's have a $\lambda = 1.3$

f. The suppressive effects of a round fired are uniform for all vehicles in the target's platoon. (subject to different λ's for different weapon systems within the same platoon).

3. Functions Represented:

   a. The effect of suppression on detection results in detections being delayed or eliminated.

   b. The effect of suppression on firing is to extend the lay/load time at target selection, and to increase aim error upon firing.

   c. The effect of unaimed direct fire is not currently implemented.

   d. The effects of suppression on unit or individual vehicle movement is not currently implemented.

4. Technique:

   a. The basic unit on which the suppression functions operate is the platoon. The suppressive of each round fired is represented by a "weight".

i.e.	120mm	APDS	weight = 1.0
	73mm	HEAT	weight = .65
	152mm	Arty	weight = 1.3

   b. The weights of all rounds fired at platoon elements are summed up and stored as attributes of the PLATOON.LEADER permanent entity. These attributes are reset every 30 seconds of the simulation.

   i.e. $r_i = r_{i-1}$ $\quad i = 1$ to 4

   where each r_i is the total suppressive weight of all rounds effecting elements of the platoon during a 30 second portion of the battle.

c. Rounds are assumed to have no suppressive effect after 2 minutes of simulation time.

d. Each r_i has associated with it a factor d_i which represents the decaying effect of suppression over time.

For example: $d_1 = 1$

$d_2 = .5$

$d_3 = .2$

$d_4 = .05$

The total effect of rounds fired at a platoon can then be represented by an adjusted value R

$$R = r_4 d_4 + r_3 d_3 + r_2 d_2 + r_1 d_1$$

$$= .05 r_4 + .2 r_3 + .5 r_2 + r_1$$

e. The suppressive effect of rounds on a particular weapon system is represented by a parameter λ .

i.e. $\lambda = 1$ for XM1's

$\lambda = 2$ for BMP's

$\lambda = 10$ for dismounted infantry

f. A time delay can then be calculated based on R and λ

$$\text{time delay } (t) = e^{R\lambda} - 1$$

This time delay value is then used to effect the functions of a particular element in the simulation.

g. The weighting of indirect fire rounds is represented as a function of proximity to the target and whether or not the impact of the round is observed by the target. The technique used is explained in a paper entitled "Suppression Methodology" (attached) and will not be further amplified herein.

5. Routines Concerned with Suppression:

a. Routine Set.Sp - reads in suppression data for each weapon system (as supplied by the user) into Array Data.Sp.

b. Routine Set.Sp - used to retrieve the appropriate suppression data from the Array Data.Sp.

c. Routine Tim.Sp - calculates the time delay (t) based on the user supplied suppression data, and the suppression related attributes of the permanent entity PLATOON.LEADER.

d. Routine Wgt.Sp.- calculates the weight of a round (relative suppressive effect), and adds that weight to the attribute ROHAT of the appropriate PLATOON.LEADER.

e. Event Reset - resets the values of the suppression related attributes of the PLATOON.LEADER permanent entity.

f. Routine Limicon - determines the probability that an impacting direct fire round is observed by the target based on the target's search sector and primary search direction.

g. Event STEP.TIME - the time it will take an observer to detect a given target is returned to event STEP.TIME by routine CARDIO. The suppressive time delay on the observer is added to this detection time. If this adjusted time is greater than the user supplied DELTA.T the detection is not allowed to take place, otherwise the detection is scheduled to occur at the adjusted time.

i. Routine Fire.Schedule - the square root of the suppressive time delay is added to the lay/load time of the firer and a firing scheduled to take place at this adjusted time.

j. Routine Geom - performs 2 functions as it applies to suppression:

1) Calls routine Wgt.Sp for rounds which hit a target or miss a target short.

2) If a minimum suppression level is met, routine geom causes the Deflection and Elevation distance of the round from its aim point to be increased by a User Supplied Multiplier (MULT.SP).

6. User Input:

User defined suppression data is read in by routine SET.SP. The user supplies values to be input into the Array Data.SP

Data.SP is dimensioned as Number of Systems by 17 (the value of Number.of.Systems is read in the MAIN prior to reading in target dimensions (TARDIM), thus there must be suppression data provided for each system for which target dimension data has been provided).

The Data.SP array is an integer array. The user inputs real or integer values (real values input by the user are converted to integers prior to being inserted into the array).

The user input for a weapon system might look like:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	50	250	30	5	1.0	.8	.1	0.	1.0	1.0	.5	.2	.05	1.1	1.2

Columns 1 & 2: Sys.Type and Wpn.Type of the Weapon System for which the data is provided.

Columns 3 & 4: Minimum and Maximum ranges (in meters) used in weighting of indirect fire rounds.

1) Within the range specified in column 3, indirect fire rounds have at least a weight of 1, this weight may increase based on the probability that the round's impact is observed.

2) Beyond the range specified in column 4, indirect fire rounds have no suppressive effect.

3) Between the two ranges specified the weight of the indirect fire round varies from 0 to 1 based on range and probability that the round's impact was observed.

Column 5: Upper bound on delay time calculation (in seconds). This value is not used in the present form of the suppression model, but was provided for further expansion of the model.

Column 6: Suppression time lower bound (in seconds) used in 2 places in the suppression portion of the model.

1) If an element's calculated time delay exceeds this lower bound at the time a round is fired, no firing stimulus detection of the firer by the element is allowed.

2) If a firer's calculated delay time exceeds this value at firing a multiplication factor (values in column 16 & 17) to increase σ_x and σ_y of the round is applied to the computed σ's in routine GEOM.

Columns 7 thru 10: Weights for rounds fired by this weapon system.

i.e. Column 7 is associated with Ammo Type 1 of the particular weapon system, column 8 with Ammo Type 2, etc.

If Sys.Type 1, Wpn.Type 2 is an XM1, then columns 7-10 above represent an individual suppressive weight of 1.0 for APDS, .8 for HEAT, .1 for 50 caliber MG, and 0 for Ammo Type 4 .

Column 11: The λ value for this weapon system. λ represents the susceptability of the system to suppression, thus the larger the λ the more "supressable" the system will be.

Columns 12 thru 15: The d_i values used in the representation of the decaying phenomenon of suppression. In the illustrated row;

$$d_1 = 1.0$$

$$d_2 = .5$$

$$d_3 = .2$$

$$d_4 = .05$$

Column 16: σ_x multiplier for increasing error in impact of a round in deflection if the firer is suppressed.

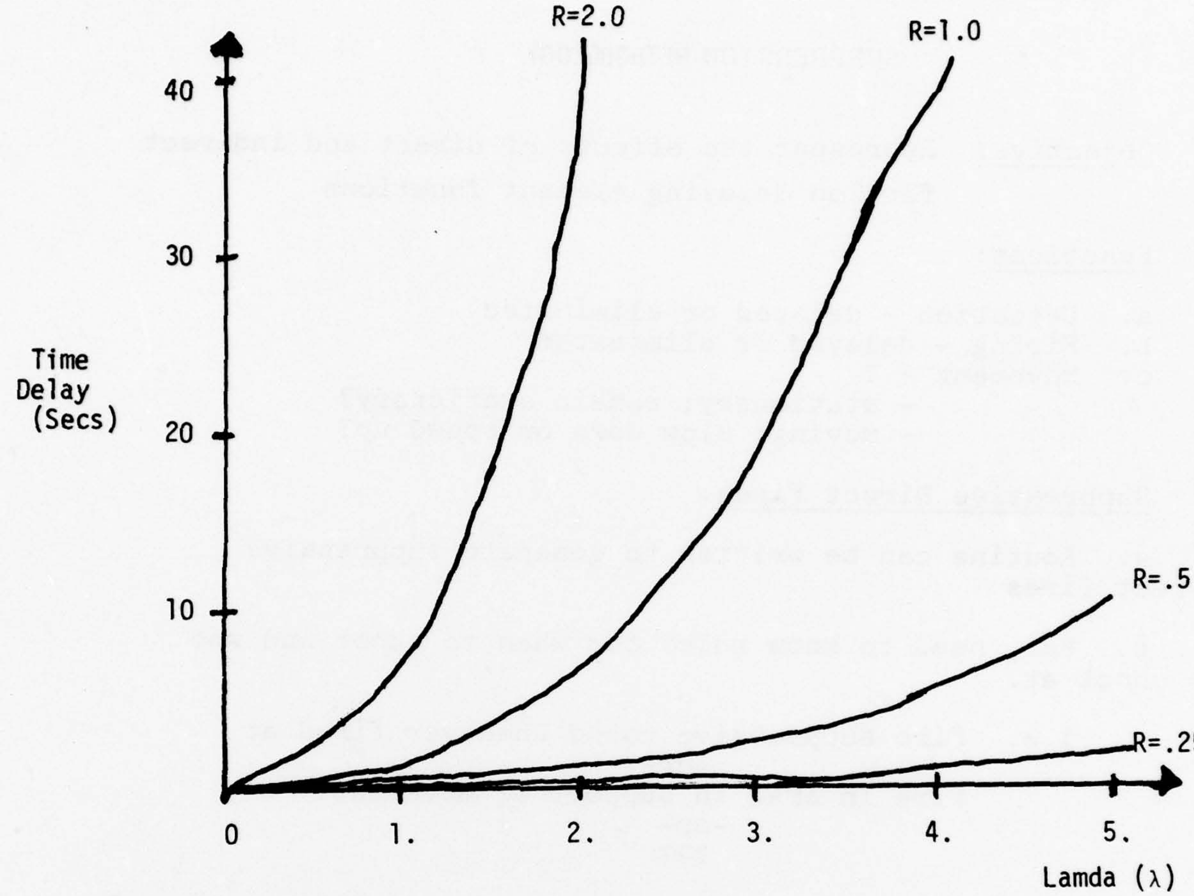
Column 17: σ_y multiplier for increasing error in impact of a round in elevation if the fired is suppressed.

7. <u>Sensitivity</u>:

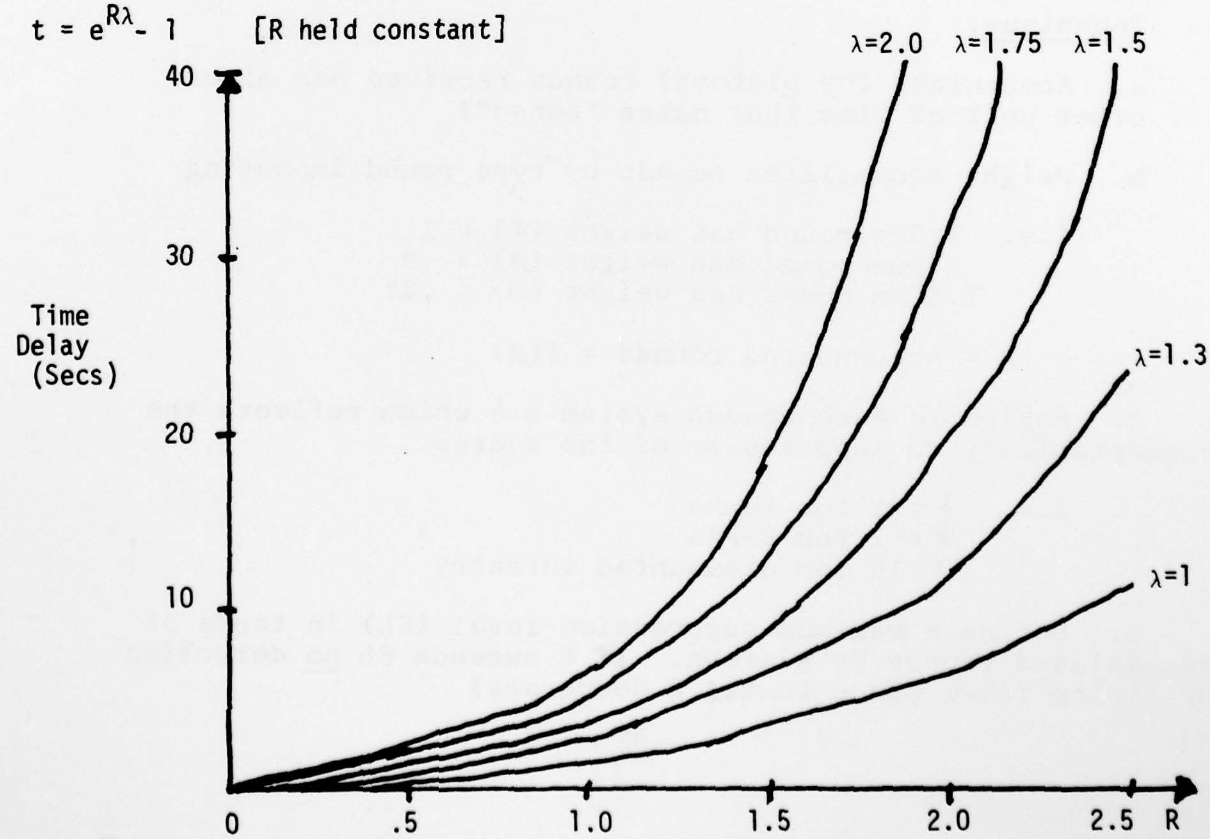
Recall that the basic formula for calculation of time delay is:

$$t = e^{R\lambda} - 1$$

To give the user some idea of the sensitivity of this time delay calculation, and to assist the user in selection of appropriate λ values for weapon systems the following graphs are provided:



$t = e^{R\lambda} - 1$ [R held constant]



$t=e^{R\lambda} -1$ (λ held constant)

SUPPRESSION METHODOLOGY

1. Objective: Represent the effects of direct and indirect fire on delaying element functions

2. Functions:

   a. Detection - delayed or eliminated
   b. Firing - delayed or eliminated
   c. Movement - ?
      - stationary; remain stationary?
      - moving; slow down or speed up?

3. Suppressive Direct Fires:

a. Routine can be written to generate suppressive direct fires

b. But, need to know rules for when to shoot and who to shoot at.

i.e. Fire suppressive round whenever fired at

-or-

Fire in area in support of movement

-or-

???

4. Technique:

a. Accumulate (by platoon) rounds received per minute (or other unit of time that makes "sense")

b. Weight accumulated rounds by type round impacting

i.e. 120mm round has weight (β) = 1
73mm round has weight (β) = .3
7.62mm round has weight (β) = .01

R = Accumulated rounds = $f(\beta)$

c. Assign to each weapon system a λ which reflects the susceptability to suppression of the system

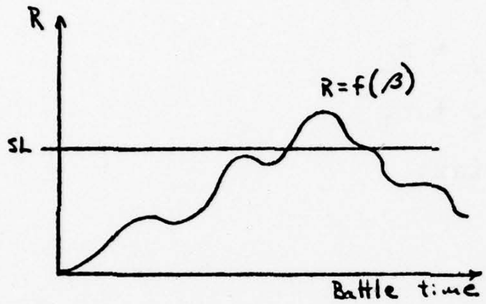
i.e. $\lambda = 1$ for tanks
$\lambda = 2$ for BMP's
$\lambda = 10$ for dismounted infantry

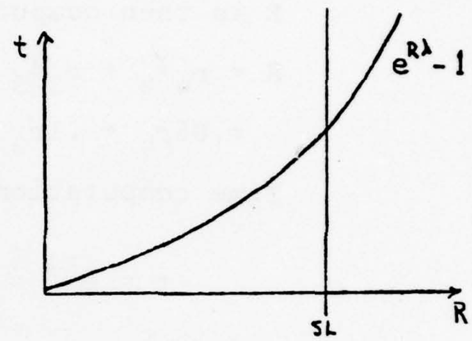
d. Define a maximum suppression level (SL) in terms of accumulated rounds by platoon. If R exceeds SL no detection or firing takes place (until R decreases)

e. Calculate time delay (t) to be added to detection times and firing times

(1) If detection time + t > some Delta.t
no detection?
-or-
allow detection to take place at adjusted time

(2) $t = e^{R\lambda} - 1$





5. <u>Calculation of R</u>:

a. R continuous: (Method 1) (uses pure accumulate)

$$R = \frac{\text{\# of rounds received}}{\text{total battle time}}$$

b. R as a weighted average: (Method 2)

(1) Every platoon has stored (for example)

rounds received per minute for;

(a) 3 minutes ago (constant)-- r_4

(b) 2 minutes ago (constant)-- r_3

(c) 1 minute ago (constant) -- r_2

(d) current rounds per minute -- r_1

A weight associated with each r_i -- γ_i

$$\left.\begin{aligned} \text{i.e. } \gamma_4 &= .05 \\ \gamma_3 &= .1 \\ \gamma_2 &= .5 \\ \gamma_1 &= 1 \end{aligned}\right\} \text{user input "guess"}$$

Each delta.t r_i are reset

i.e. $r_i = r_{i-1}$

r_1 continues to be accumulated as a continuous value

R is then computed as follows:

$$R = r_4\gamma_4 + r_3\gamma_3 + r_2\gamma_2 + r_1$$
$$= .05r_4 + .1r_3 + .5r_2 + r_1$$

Time computation remains:

$$t = e^{R\lambda} - 1$$

c. R as a weighted average: (Method 3)

$$\hat{R} = \sum_{i=1}^{\infty} w_i x_i \qquad w_1 \quad w_2 \quad \ldots w_i$$

use α as a smoothing constant and use geometric distribution for weights

$$R = \sum_{i=1}^{\infty} (1-\alpha)^{i-1} \alpha x_i$$

where x_i are R values from previous step.time increments

For example (if we truncate suppression effects to the latest 4 step.time increments)

r_4 = rounds received 3 mins ago

r_3 = rounds received 2 mins ago

r_2 = rounds received 1 min ago

r_1 = current rounds per minute count

Then:

$$R = \sum_{i=1}^{4} (1-\alpha)^{i-1} \quad r_i$$

time computation remains: $t = e^{R\lambda} - 1$

d. R as a moving average (Method 4)

$$\hat{R} = \sum_{i=1}^{j} r_i$$

For example let j = 4

Define r_1, r_2, r_3, and r_4 as before

Then:

$$\hat{R} = \sum_{i=1}^{4} r_i$$

$$= r_1 + r_2 + r_3 + r_4$$

at each step.time reset r_i's where $r_i = r_{i-1}$

time computation remains: $t = e^{R\lambda} - 1$

6. Comments:

a. Direct and indirect suppression can both be represented as a single function

b. Effects of suppression played parametrically based on user input values for β, λ, SL, γ.

c. Easily implemented using SIMSCRIPT accumulate feature

d. Suppression can be turned off by setting $\beta = 0$ for all rounds

e. Suppression effects different for different systems through the use of the λ attribute.

f. "Rules" needed before suppressive direct fires can be played

g. Suppression effects on movement are still unclear.

INDIRECT FIRE SUPPRESSION

1. <u>Purpose</u> : Offered as a possible extension of the proposed suppression model

2. <u>Assumptions</u>:

a. There exists a minimum range within which there is a uniform suppressive effect

b. There exists a maximum range beyond which no suppression takes place

c. Between min and max ranges suppressive effects are a function of;

1) Target to impact range

2) Probability that the impact is "observed" by target

3. <u>Technique</u>:

a. Accumulation of rounds as per general suppression model.

b. Time delay $t = e^{R\lambda} - 1$

c. Weight type of round impacting by some parameter

i.e. 8" round $\beta = 1.5$

155mm round $\beta = 1.0$

105mm round $\beta = .8$

d. Weighted round values are accumulated by platoon

e. Use Lemicon distribution to determine probability that the impact of the round is observed

f. Normalize the Lemicon value such that a round impacting on direct line with the target's primary search direction has a weight of 1.

g. Adjust the weight based on the range from target to point of impact

i.e.

weight = (min range/actual range)^x.p.sub.k

where: min range is user input

actual range is calculated

p.sub.k is normalized Lemicon value

x is a range adjustment factor (set equal to 1 until test data is available)

h. min range/actual range has max value of 1

i. p.sub.k ahs max value of 1

j. Equation for calculation of round weights then becomes:

weight = 0 if range > max range

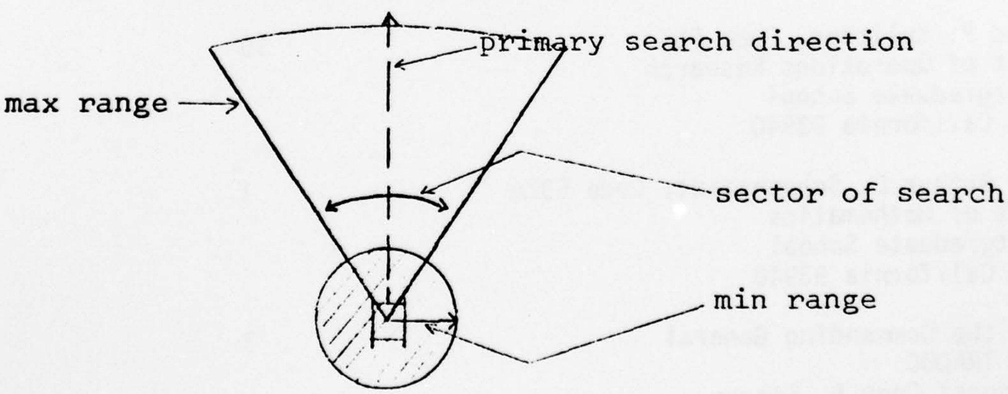
$$\text{weight} = \beta \left\{ \left[(\text{min range/actual range})^{x} . \text{p.sub.k} \right] + \delta \right\}$$

where β is weight for type of round

δ = 0 if range > min range

δ = 1 if range $\leq$ min range

h. Graphically:



within shaded area: weight =(1 + Lemicon value).β

max value is 2β

external to shaded area but within wedge

weight = (Lemicon value).β, max value is β

weight = 0 elsewhere

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Professor James K. Hartman Code 55Hh Department of Operations Research Naval Postgraduate School Monterey, California 93940	10
5.	Professor S. H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93940	10
6.	LTC Edward P. Kelleher, Code 55Ka Department of Operations Research Naval Postgraduate School Monterey, California 93940	10
7.	Professor Arthur L. Schoenstadt, Code 53Zh Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
8.	Office of the Commanding General U.S. Army TRADOC Attn: General Donn A. Starry Ft. Monroe, Virginia 23651	1
9.	Headquarters U.S. Army Training & Doctrine Command Attn: ATCG-T (Col. Ed Scribner) Ft. Monroe, Virginia 23651	1

10. Headquarters
U.S. Army Training & Doctrine Command 1
Attn: Director, Analysis Directorate
Combat Developments (MAJ Chris Needels)
Ft. Monroe, Virginia 23651

11. Headquarters 1
U.S. Army Training & Doctrine Command
Attn: Director, Maneuver Directorate
Combat Developments (Col. Fred Franks)
Ft. Monroe, Virginia 23651

12. Mr. David Hardison 1
Deputy Under Secretary of the Army
(Operations Research)
Department of the Army, The Pentagon
Washington, D.C. 20310

13. LTG William Richardson 1
Commanding General
U.S. Army Combined Arms Center
Ft. Leavenworth, Kansas 66027

14. Director 1
Combined Arms Combat Development Activity
Attn: Col. Reed
Ft. Leavenworth, Kansas 66027

15. Director, BSSD 1
Combined Arms Training Development Activity
Attn: ATZLCA-DS
Ft. Leavenworth, Kansas 66027

16. Director 1
Combat Analysis Office
Attn: Mr. Kent Pickett
U.S. Army Combined Arms Center
Ft. Leavenworth, Kansas 66027

17. Command and General Staff College 1
Attn: Education Advisor
Room 123, Bell Hall
Ft. Leavenworth, Kansas 66027

18. Dr. Wilbur Payne, Director 1
U.S. Army TRADOC Systems Analysis Activity
White Sands Missile Range, New Mexico 88002

19. Headquarters, Department of the Army 1
Office of the Deputy Chief of Staff
for Operations and Plans
Attn: DAMO-2D
Washington, D.C. 20310

20. Commander 1
U.S. Army Concepts Analysis Agency
8120 Woodmont Avenue
Attn: MOCA-SMS (CPT Steve Shupack)
Bethesda, Maryland 20014

21. Commander 1
U.S. Army Concepts Analysis Agency
Attn: LTC Earl Darden-MOCA-WG
8120 Woodmont Avenue
Bethesda, MD. 20014

22. Director 1
U.S. Army Night Vision & Electro-optical Lab.
Attn: DEL-NV-VI-Mr. Bob Hermes
Fort Belvoir, VA 22060

23. Director 1
U.S. Army Material Systems Analysis Activity
Attn: Mr. Will Brooks
Aberdeen Proving Grounds, Maryland 21005

24. Director 1
Combat Developments Experimentation Command
Attn: CPT William J. Caldwell
Fort Ord, California 93941

25. Director 1
Armored Combat Vehicle Technology Program
Attn: Col. Fitzmorris
U.S. Army Armor Center
Ft. Knox, Kentucky 40121

26. Col. Frank Day 1
TRADOC Systems Manager - XM1
U.S. Army Armor Center
Ft. Knox, Kentucky 40121

27. Director 1
Combat Developments, Studies Division
Attn: MAJ W. Scott Wallace
U.S. Army Armor Agency
Fort Knox, KY 40121

28. Commandant 1
U.S. Army Field Artillery School
Attn: ATSF-MBT (CPT Steve Starner)
Fort Sill, Oklahoma 73503

29. Director 1
Combat Developments
Attn: Col. Clark Burnett
U.S. Army Aviation Agency
Fort Rucker, Alabama

30. Director 1
Combat Developments
U.S. Army Infantry Agency
Fort Benning, GA

31. Director 1
Missile Intelligence Agency
Attn: ADA Tactics (CPT E.G. Hagewood)
Redstone Arsenal, AL 35809

32. Director 1
Combat Developments
ATTN: CPT William D. Meiers
U.S. Army Air Defense Agency
Fort Bliss, TX 79905

33. Commander 1
U.S. Army Logistics Center
ATTN: ATCL-OS-Mr. Cammeron/CPT Schuessler
Fort Lee, VA 23801

34. Commander, USAMMCS 1
ATTN: ATSK-CD-CS-Mr. Lee/Mr. Marmon
Redstone Arsenal, AL 35809

35. Commander 1
U.S. Army Combined Arms Center
ATTN: ATZL-CA-CAT (R.E. DeKinder, Jr.)
Fort Leavenworth, KA 66027

36. Director 1
U.S. Army AMSAA
ATTN: DRXSY-AA (Mr. Tom Coyle)
Aberdeen Proving Grounds, MD 21005

37. R. Stampfel, Code 55 1
Department of Operations Research
Naval Postgraduate School
Monterey, California 93940

38. Dean of Research 1
Naval Postgraduate School
Monterey, Ca 93940